Simplicial Methods for Solving Selected Problems in General Relativity Numerically

REGGE CALCULUS AND THE FINITE-ELEMENT METHOD

DISSERTATION

zur Erlangung des akademischen Grades

doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der Fakultät für Mathematik und Informatik
der Friedrich-Schiller-Universität Jena

von Dipl.-Phys. Frank Peuker
geboren am 16.Juli 1979 in Schwedt/Oder

Gutachter

1. Prof. Dr. Gerhard ZUMBUSCH
Lehrstuhl für Wissenschaftliches Rechnen – Institut für Angewandte Mathematik – Fakultät für Mathematik und Informatik – Friedrich-Schiller-Universität Jena

2. Prof. Dr. Bernd BRÜGMANN
Numerische Relativitätstheorie – Theoretisch-Physikalisches Institut – Physikalisch-Astronomische Fakultät – Friedrich-Schiller-Universität Jena

Referees

1st: Prof. Dr. Gerhard ZUMBUSCH
Scientific Computing Group – Institute of Applied Mathematics – Department of Mathematics and Computer Science – University of Jena

2nd: Prof. Dr. Bernd BRÜGMANN
Numerical Relativity Group – Institute of Theoretical Physics – Faculty for Physics and Astronomy – University of Jena

Tag der öffentlichen Verteidigung:

1. September 2009

Zusammenfassung

In der vorliegenden Arbeit werden zwei numerische Verfahren betrachtet, welche spezielle Probleme der Allgemeine Relativitätstheorie näherungsweise berechnen können. Dies ist zum einen die Finite-Element-Methode und zum anderen das Regge-Kalkül. Beide Verfahren basieren auf einer Zerlegung des betrachteten Gebietes in Simplizes. Zahlreiche dieser Simplizialzerlegungen und ihre Anwendbarkeit auf beide Verfahren wurden in dieser Arbeit eingehend untersucht bevor diese zur Lösung des Problems verwendet wurden.

Um Probleme aus der Allgemeinen Relativitätstheorie numerisch zu berechnen, wird in dieser Arbeit die 3+1-Zerlegung angewendet. Diese unterteilt die Lösung des Problems in zwei Teilschritte. Im ersten Schritt werden Anfangsdaten bestimmt mit Hilfe derer man im zweiten Schritt ein Zeitentwicklungsschema anwenden kann. Der erste Teil dieser Arbeit demonstriert, wie man das Anfangsdaten-Problem für spezielle Probleme unter Verwendung der Finiten-Element-Methode lösen kann. Der zweite Teil der Arbeit widmet sich der Aufgabe, Anfangsdaten mit Hilfe des Regge-Kalküls zu entwickeln. Während viele Formulierungen in der Allgemeinen Relativitätstheorie koordinatenabhängig sind und somit eine Vielzahl an Lösungen ein und dasselbe Problem beschreiben, verwendet das Regge-Kalkül Kantenlängenquadrate als Variablen. Die Länge einer Kante ist unabhängig von dem zugrundeliegenden Koordinatensystem. Die Lösung eines Problems wird somit durch genau einen Satz an Kantenlängenquadraten repräsentiert. Dass man auf das Quadrat dieser Länge zugreift, liegt daran, dass die vierdimensionale Raumzeit nicht euklidisch ist, sondern Minkowski-Signatur besitzt. Abhängig vom Vorzeichen des Kantenlängenquadrates ergibt sich eine räumliche Ausdehnung oder eine Zeitdifferenz.

Anhand vieler Beispiele werden beide Verfahren untersucht. Für die Anfangsdaten wurde ein statisches und ein sich rotierendes schwarzes Loch sowie zwei sich aufeinander zubewegende schwarze Löcher betrachtet. Für die Zeitentwicklung wurden Beispiele der Apples-With-Apples-Testsuite, das Kasner-Universum und ein statisches schwarzes Loch untersucht. Es zeigt sich, das beide Verfahren auf einfache Probleme der Numerischen Relativitätstheorie anwendbar sind. Die Finite-Element-Methode liefert Anfangsdaten auf einer Excision-Domäne, wobei hier einer selbst konstruierten Triangulierung einer unstrukturierten der Vorzug zu geben ist.

Im Regge-Kalkül ist es erstmals gelungen, unstrukturierte Gitter, welche von externen Gittergeneratoren erstellt werden, als Grundlage für Zeitentwicklungen im Regge-Kalkül zu benutzen. Damit können auch komplexe Gebiete schnell modelliert werden. Des weiteren zeigen die Ergebnisse dieser Arbeit, dass das QR-Verfahren einem sonst üblichen LU-Verfahren vorzuziehen ist. Vor allem in fast flachen Raumzeiten und bei der Zeitentwicklung von unstrukturierten Gittern zeigt sich, das erst durch die Anwendung des QR-Verfahrens eine stabile Simulation möglich ist. Durch den Verzicht auf die Anwendung einer simplizialen Bianchi-Identität gelingt es entsprechende Probleme zu lösen und die Ergebnisse mit bestehenden Resultaten aus Finite-Differenzen-Verfahren zu vergleichen. Erstmals wurde die Konvergenz des Regge-Kalküls anhand integraler Normen der Metrik untersucht. Bisher wurden nur Abweichungen von Kantenlängen diskutiert oder das

Residuum der Regge-Gleichungen betrachtet. Mittels des Linear-Wellen-Tests aus der Apples-With-Apples-Testsammlung wird das Verhalten der numerischen Lösung in den neuen Normen eingehend untersucht. Hierbei wurde auch eine aus der Kausalität abgeleitete Courant-Friedrichs-Levi-Bedingung (CFL-Bedingung) numerisch bestätigt.

Simulationen des Kasner-Unviersums und des Gowdy-Universums zeigen, dass analytisch bekannte Lösungen qualitativ und quantitativ erfolgreich numerisch approximiert werden können. Im ersten Fall stimmt die zeitentwickelte Lösung mit der analytischen bis auf einen relativen Fehler in der Größenordnung von 10^{-5} überein. Auch im Kasner-Universum wird die theoretisch ermittelte CFL-Bedingung numerisch bestätigt. Zudem ist es auch erstmals gelungen, Probleme auf Domänen mit räumlichen Rand zu lösen, wobei geeignete Bedingungen an Randkanten ausschlaggebend für die Stabilität sind. Hiermit wurden auf unstrukturierten Gittern die gestörte flache Raumzeit und die Schwarzschild-Raumzeit zeitentwickelt. Es zeigt sich, das mit besseren Winkeln des Gitters auch die Zeitentwicklung stabiler wird.

Das Regge-Kalkül wie auch die Finite-Element-Methode sind vielversprechende und wie in dieser Arbeit gezeigt wurde funktionierende Lösungsansätze für einfache Probleme der Numerischen Relativitätstheorie sind. Durch die höhere Flexibilität dieser Simplizialmethoden, sind sie interessante Alternativen zu bisherigen, auf Finiten Differenzen basierenden Ansätzen. Weiterführende Arbeiten auf diesem Gebiet könnten beide Methoden dahingehend weiterentwickeln, komplexere Probleme, wie das Einspiralen zweier schwarzer Löcher, zu lösen. Solche alternativen Lösungen könnten bestehende Verfahren verifizieren und erweitern.

Contents

CHAPTER 1

Introduction

The present work assumes basic knowledge in the field of Mathematics and General Relativity. The next section introduces basic ideas, formulas and name conventions. A deeper understanding can only be gained by means of additional literature. In my view, I found the book [Car03] from Sean CARROLL and the compendium [MTW73] of MISNER, THORNE and WHEELER outstanding. The introduction is also based on [Rus45] and [Reg].

1.1 Gravitation and General Relativity

For a long time myths or religions gave the best explanations. Then, THALES OF MILET successfully forecasted a solar eclipse in 585 B.C., proving that it is possible to make predictions of future astronomical events by examining the past. After the middle ages the interest in systematic exploration of the laws of nature arose : Observational data lead to mathematical relations which explain nature. Tycho BRAHE collected a lot of data on the subject of astronomical objects with very precise measurements. By means of this data, his assistant Johannes KEPLER discovered *Kepler's laws*. To explain the data of BRAHE he had to break with the ancient ideal of circular orbits, replacing them with elliptical ones. Meanwhile, Galileo GALILEI explained the complicated and mysterious parallax trajectories with *heliocentrism*: The earth moves around a fixed sun. Also, he confirmed that any two different bodies fall with the same acceleration and an object on which no force is acting travels on a linear trajectory. Combining *Kepler's laws* and the kinematic laws of GALILEI, one of the most important achievements of Isaac NEWTON was the *Newton's law of universal gravitation*:

$$F = \frac{G \cdot M \cdot m_g}{r^2} \,. \tag{1.1}$$

Here, F is the force acting on the object of mass m_g. From special observational data and exceptional insight a universal law was found which stated a force between any two objects only depending on the gravitational mass M and m_g of each body and the distance r of its separation. So, the force which let astronomical objects move is the same that let things fall on earth. The theorem was a great success and it took more than 200 years until a more accurate theory was needed. One key observation was the equivalence of inertial mass m_i and gravitational mass m_g measured with high accuracy by Lorand EÖTVÖS [1] in 1906. Given this, the mass of the object on which the force is acting drops out from the dynamic equation $m_i a = G m_g M/r^2$, leading to $a = GM/r^2$. Because the properties of the object do not influence the dynamics at all, it is possible to have a geometrical view on gravitation: Matter as the source of gravitation does not influence an object via an long-range gravitational force but changes geometry itself. Then, any test body on which no additional force is acting moves along natural paths in the now curved geometry. These natural paths are called geodesics and are the generalized concept of straight lines in flat geometry. They are the shortest distance between two points with respect to a non-euclidean metric. The idea that matter influences geometry is an improvement as it reveals pseudo forces in Newton's theory for example. This is well illustrated with two observers on the 2-sphere of the earth. If they start from different positions at the equator and move to the north pole on geodesics, than their relative distance will decrease and the two trajectories will cross at the north pole. After passing it, the relative distance will increase till they reach the equator again. On long time scales, the relative distance is oscillating. As they travel along parallel trajectories, in Newton's theory a pseudo force had to be introduced to give a theory of the right dynamics. This pseudo force vanishes in curved space, where parallel trajectories do not need to have the same relative distance.

Albert EINSTEIN developed the geometrical theory of gravitation in 1915: *The General Theory of Relativity*. It works on a four-dimensional manifold with three spatial and one time dimension. In this way, the formerly always separately treated space and time are summarized in a four-dimensional spacetime. It has Minkowski signature (also called Lorentzian signature) which means that the corresponding components of space and time in the scalar product get opposite signs. In absence of matter, the squared distance l^2 of $A = (t_0, x_0, y_0, z_0)$ and point $B = (t_1, x_1, y_1, z_1)$ would be calculated as:

$$l^2 = -(t_1 - t_0)^2 + (x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2 \,. \tag{1.2}$$

Eq.(1.2) reveals that not only positive l^2 values are possible, but also negative and zero values. If $l^2 > 0$, the connecting line is called spacelike. If $l^2 = 0$ for different points, then the line is called lightlike and if $l^2 < 0$, the line is called timelike. Fig.(1.1) gives a short sketch of these different types of lines and it shows the light cone.

Given a special point P in spacetime, the light cone is the surface formed by all lines through one given point which have zero edge length. It is an important surface to discuss causality, that means to find an answer to the question: Which domain of spacetime can influence a special event in the future?

To work in curved spacetime makes the introduction of new values necessary. The 4-dimensional manifold is completely described with the metric $g_{\mu\nu}$ of Minkowski signature. Greek indices in general, and here μ and ν, range from 0 to 3. Sometimes it is useful to consider only spatial indices. Then Latin indices, ranging from 1 to 3 will be used. The metric is symmetric and so the number of independent components drops to ten. In curved spacetime, the infinitesimal distance is given by,

$$\mathrm{d}s^2 = \sum_{\mu\nu} g_{\mu\nu} \mathrm{d}x^\mu \mathrm{d}x^\nu \;=:\; g_{\mu\nu} \mathrm{d}x^\mu \mathrm{d}x^\nu \,. \tag{1.3}$$

---

[1] This experiment directly influenced A. Einstein. Later experiments measured the equality for higher accuracies. For a recent improvement in accuracy see [SCA$^+$08]. The fact is known as *weak equivalence principle*.

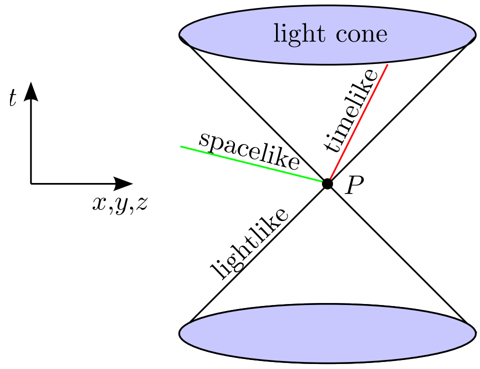


Fig. 1.1: Light cone to a given point P. Timelike line (red), spacelike line (green) and a lightlike line are shown. Causality: The upper light cone is the region of spacetime which is influenced by point P. The lower light cone is the region of spacetime which influences point P.

Any finite distance must now be determined by evaluating an integral, since each $g_{\mu\nu}$ is a function of spacetime. Also, the so called *Einstein summation convention* admits the supression of the summation symbol with the rule that if an index occurs in alternating sub- and superscript position in two factors of a product, then this index must be summed over, eliminating this index in the result. With a metric-compatible connection (Christoffel symbols)

$$\Gamma^{\gamma}_{\alpha\beta} = \frac{1}{2} g^{\gamma\delta}(\partial_\alpha g_{\beta\delta} + \partial_\beta g_{\alpha\delta} - \partial_\delta g_{\alpha\beta}) \tag{1.4}$$

the curvature at any point can now be expressed through the *Riemann tensor*. It is defined as

$$R^{\alpha}{}_{\beta\gamma\delta} = \partial_\gamma \Gamma^{\alpha}_{\delta\beta} - \partial_\delta \Gamma^{\alpha}_{\gamma\beta} + \Gamma^{\alpha}_{\gamma\rho}\Gamma^{\rho}_{\delta\beta} - \Gamma^{\alpha}_{\delta\rho}\Gamma^{\rho}_{\gamma\beta}\,. \tag{1.5}$$

The Riemann tensor applied to any vector expresses how much this vector is changed when parallel transported around a given closed infinitesimal loop.

The source of gravitation is the distribution of matter and how it moves. This is summarized in the *energy momentum tensor* $T_{\alpha\beta}$. Fig.(1.2) gives an idea in a two-dimensional sketch of how matter is supposed to influence geometry.

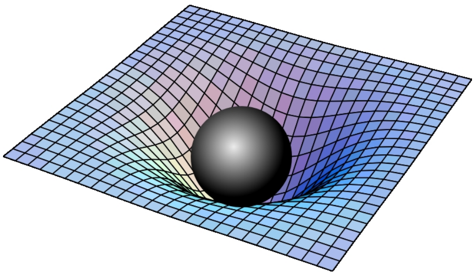


Fig. 1.2: Matter acting on 2d curvature. 2d abstraction for the idea how a matter distribution influences geometry.

When trying to state an equation between the sources of gravitation and the curvature, it turns out, that the Riemann tensor is not the object of choice but a so called *contracted* version from it. This means, components of a tensor are summed over. This becomes clear in notation and regarding the Einstein summation convention. The first contracted version is called *Ricci tensor* and is defined as $R_{\beta\delta} = R^{\alpha}{}_{\beta\alpha\delta}$. The next contraction is called *Ricci scalar* and is $R = R^{\alpha}{}_{\alpha}$. Now, the Einstein field equations can be written in the form

$$R_{\alpha\beta} - \frac{1}{2} R g_{\alpha\beta} = 8\pi T_{\alpha\beta}\,. \tag{1.6}$$

In this equation *geometrized units* were used. This means the speed of light c and the gravitational constant G from Eq.(1.1) is set equal to one: $c = G = 1$.
Another concept of General Relativity not mentioned yet is locality. If we look at a sufficiently small region of spacetime, it looks approximately flat. In an infinitesimal neighborhood of a point General Relativity can be formulated as Special Relativity. Then, the observer would not feel any graivitational field. Only on finite scales an observer would see new forces arising due to spacetime curvature, called *tidal forces.* Upon choosing a non-inertial system additional pseudo forces come into play. A further aspect of General Relativity is an upper limit of velocities between inertial systems. Contrary to Newton's theory, where gravitation is acting instantly over a long distance, General Relativity restrict the velocity of signal transfer to the speed of light.

To a certain extent the theories of MAXWELL and EINSTEIN can be compared: The electrostatics correspond to Newton's gravitation whereas electrodynamics correspond to General Relativity. An immediate outcome of General Relativity is for example the existence of *gravitational waves.* Newton's theory does not admit those.

In this work two different formulations of the Einstein equations are applied. To derive these, it is necessary to apply a Hamilton principle, that means to state an action which is then minimized with respect to a special set of variables. Different sets of variables give different systems of equations, but all systems describe the same problem. In General Relativity, this action is called *Einstein-Hilbert action.* It takes the simple form:

$$S = \int R \, \mathrm{d}V\,. \tag{1.7}$$

$\mathrm{d}V$ is the four-dimensional curved infinitesimal volume element and the integral covers the whole domain of spacetime on which the problem is posed.

1.2 Outline

This work is divided into two different parts. The first part deals with initial data determined with the Finite-Element method. In chapter two, from the formulation from ARNOWITT, DESER and MISNER (ADM formalism) the initial data problem is derived and applied together with the Finite-Element method (FEM). The ADM formalism splits up spacetime by means of a 3+1 decomposition of spacetime. In this way the problem is separated in a three-dimensional initial data problem and a four-dimensional Time-Evolution problem. Initial data for one and two black holes are determined using a modified version of the external developed code `KASKADE`. These modifications were done by myself and allow to solve semilinear partial differential equations instead of linear partial differential equations only.

In the second part of this thesis, Time-Evolution problems are solved with Regge Calculus. In section 3.1 this completely coordinate-free approach for an approximated spacetime is presented. General Relativity emphasizes the uniqueness of geometrical objects and the ambiguity of a formulation in coordinates. Contrary to Regge Calculus, the Einstein equations are formulated on the background of a chosen coordinate system. Many solutions of the Einstein equations can describe one and the same problem, only expressed in different coordinates. In Regge Calculus a solution on one chosen triangulation is unique. In section 3.2 different triangulation schemes are discussed with respect to the applicability with the Time-Evolution scheme. Section 3.3 presents the algorithms developed and implemented in a C++ program within this thesis. In section 3.4 several improvements investigated within this thesis are shown by means of many test problems.

First, a two-dimensional problem works as a preceeding testbed to show feasibility of the Time-Evolution scheme. From there on, fully four-dimensional problems without any posed symmetries are treated. The impact of a spatial boundary to the Time-Evolution scheme is investigated and boundary conditions are identified to make the evolution of domains with a spatial boundary possible. By means of the Schwarzschild spacetime it is illustrated how boundary conditions affect the stability of these evolutions.

For the first time, it is shown that a QR scheme makes Time-Evolutions of almost flat spacetimes and the evolution of data on unstructured grids just possible. A LU scheme fails in the sense that the error growths much faster and the simulation breaks down much earlier compared to the QR scheme which often does not break down at all. Also, a new view on error growth is presented in this work. Instead of regarding edge length or residuals only, the 3-metric is reconstructed and integral error norms from Finite-Element theory were applied to the 3-metric for the first time. With this error measures the convergence of Regge Calculus in the Linear-Wave testbed is investigated. The discretization error shows a linear convergence with respect to the spatial discretization length in infinity and L_2 norm whether the error growth in time shows a quadratic convergence in both norms. Furthermore a Courant condition is identified which match exactly the theoretical prediction from causality.

In the next two concluding chapters signficantly non-linear spacetimes were investigated. By means of the Kasner universe and Gowdy universe it is shown how lapse and shift influence the Time-Evolution. In the case of the Kasner universe, analytical edge length and calculated edge length does only deviate relatively with a factor of 10^{-5}. In both universes, the calculated edge length and calculated metric components respectively show qualitatively correct behaviour. At last Schwarzschild spacetime is evolved with an unstructured grid generated with external software. This was done for the first time in this setting and it turned out that the unstructured grid triangulating the hypersurface influences the Time-Evolution in the sense that a small maximum angle and a high minimum angle respectively make the evolution more stable.

Both methods, the Finite-Element method and Regge Calculus, are employed on a simplicial decomposition of the domain of spacetime. Before applying each method, different decomposition methods are discussed regarding applicability in each scheme. Two triangulations were developed as part of this thesis. First, the structured mesh `MESHGEN` was constructed to cover the excision domain. Second, an algorithm was found to construct four-dimensional grids with respect to arbitrary three-dimensional grids. The generated grids make use of the prism method and are then compatible with the Time-Evolution scheme.

1.3 Motivation

As a completely new outcome of Einstein theory, gravitational waves are predicted. It is from very high interest for the scientific community to detect this new phenomena. Unfortunately, the amplitude of such a wave observed on earth or in the orbit of earth is very small. Only recent modern technology put gravitational wave detectors like GEO-600, LIGO or VIRGO to an accuracy which is needed to detect gravitational waves. The data collected with terrestrical detectors is analyzed by statistical methods to determine a gravitational wave within the noise signal. For this statistical approach the theoretical waveform must be avaiable. Another approach to detect gravitational waves with less noise is the LISA project (Laser Interferometer Space Antenna project) which is expected to launch in 2018. It's presuming LISA Pathfinder Mission is scheduled for the end of 2009. This extra-terrestrical experiment is supposed to detect gravitational waves much more accurately.

Waveforms can be calculated analytically with Einsteins equations. Since the publication of Einsteins theory, a lot of analytical solutions have been found. But due to the very high complexity they are often based on symmetry assumptions and simplifications. Unfortunately, gravitational waves are not emitted by simple static objects but need a dynamic setting. One favoured setting which gives a relative high amplitude in the signal is the binary black hole inspiral (BBH inspiral). Here, two black holes which orbit each other are losing energy due to gravitational waves. When they are close to each other they merge and give one black hole in the end. The calculation of the waveform for such an event is currently out of scope for analytical approaches. It is too complex. Instead, approximate solutions are calculated. By means of computers and numerical mathematics these can be computed, but to find a working setting is also challenging. Only recently the calculation of a BBH inspiral became possible. Up-to-date codes facilitate almost exclusively Finite Differences as the discretization scheme. Simplicial methods like the Finite-Element method or Regge Calculus does play only a minor role. But it can be expected that these approaches give better results to some extent since they are more flexible and appproximate the geometry better. This work is motivated to have a closer look on these both simplicial methods: Are they able to challenge existing codes? Can they calulate relevant problems of General Relativity? Is it possible to develop a simplicial method this far that it can contribute to the calculation of gravitational waveforms? With such alternative methods results from existing codes could be verified or maybe extended.

The existence of gravitational waves is out of doubt. HULSE and TAYLOR compared in 1975 a spin-down of the neutron star system PSR1913+16 with the prediction from General Relativity. They found a great agreement of the energy-loss and earned the nobel prize for this discovery. The next step is to observe gravitational waves directly supported by theoretical waveforms from numerical approaches. To investigate abilities of simplicial methods which could verify or extend existing results is the motivation for this work.

CHAPTER 2

The Finite-Element Method With The Initial-Data Problem

For solving problems in General Relativity, spacetime is splitted up in space and time which is known as $3+1$ decomposition. The space is represented with a three-dimensional manifold, this is a *hypersurface* of spacetime. Each hypersurface can be labeled with a time coordinate. In the infinitesimal limit, infinite many hypersurfaces cover the spacetime completely. This is called *foliation* of spacetime. Many foliations are possible. If we choose one foliation which seems suitable to solve the problem we can employ a two-step approach. Given a special region of spacetime on which the problem should be solved, we start with the first hypersurface and calculate the problem on it. This is called *initial data problem.* Then, we can carry this initial solution to the next hypersurface and repeat this to get a solution for the whole domain. This is called *Time-Evolution problem.*

In the first section, we will derive a semilinear elliptical differential equation by means of the formalism of Arnowitt, Deser and Misner (ADM equations) and further following the approach of Bowen and York in [BJ80]. In the second section it is discussed how this equation is solved with the Finite-Element method (FEM). I will use the FEM code package `KASKADE` which usually solves linear elliptical differential equations only. With a medium amount of manipulation done within this thesis this code is also applicable to the initial data problem. The third section will deal with simplicial decompositions, also called *triangulations.* This is an important issue in the FEM. In general, different triangulations yield different good results concerning error, convergence and speed of convergence. The last section presents the results obtained with the modified `KASKADE` code. One static black hole, one black hole with linear momentum and two black holes with linear momentum in a *head-on collision* setting are discussed.

2.1 Theoretical Foundations of the Initial Data Problem

First, the aim is to separate Einsteins equations in dynamic equations, necessary for Time-Evolution, and constraint equations, necessary for the initial data problem. Then, the four constraint equations can again be separated in three *momentum constraint equations* and one *Hamiltonian constraint equation.* The first three equations can be solved in general as shown in [BJ80]. They give rise to a function which can then be used in the last equation.

2.1.1 The Einstein-Hilbert Action and Einstein Equations

As the kinematic laws in classical mechanics can be expressed with an action integral, in General Relativity the Einstein equations can be expressed with the *Einstein-Hilbert action.* This *Hamilton principle* can be regarded to be the starting point for constructing partial differential equations which solve a problem of General Relativity. With R the Ricci Scalar, $\mathrm{d}V$ the four-dimensional volume element and Ω the domain of the problem, the Einstein-Hilbert action is defined as

$$S = \int_\Omega R \, \mathrm{d}V \, . \tag{2.1}$$

To derive equations which represent a solution to the problem, this action must be extremized, that means $\partial S = 0$ is demanded. To do this, a special set of values must be chosen. This set of variables must be complete and suppress any degrees of freedom. Now, the variation can be applied explicitly. This gives us either differential equations if the value of the variation is a function, or algebraic equations if the values of variation are ordinary variables.

The Einstein equations are derived by varying this action with respect to the 4-metric $g_{\mu\nu}$, which is a function of spacetime. At first glance, the Einstein equations Eq.(1.6) look almost easy. This is because the complexity is hidden in the Ricci Scalar R and the Ricci Tensor $R_{\mu\nu}$. They depend implicitly on $g_{\mu\nu}$. If we make the dependence on $g_{\mu\nu}$ explicit we get a much more complicated formulation:

$$\Gamma^\alpha_{\beta\gamma} = \frac{1}{2} g^{\alpha\delta} \left(\frac{\partial g_{\delta\beta}}{\partial x^\gamma} + \frac{\partial g_{\delta\gamma}}{\partial x^\beta} - \frac{\partial g_{\gamma\beta}}{\partial x^\delta} \right) , \tag{2.2}$$

$$\begin{aligned} 8\pi T_{\alpha\beta} = & \frac{\partial \Gamma^\delta_{\alpha\beta}}{\partial x^\delta} - \frac{\partial \Gamma^\delta_{\delta\beta}}{\partial x^\alpha} + \Gamma^\delta_{\alpha\beta} \Gamma^\gamma_{\delta\gamma} - \Gamma^\delta_{\gamma\beta} \Gamma^\gamma_{\delta\alpha} \\ & - \frac{1}{2} g_{\alpha\beta} g^{\delta\gamma} \left(\frac{\partial \Gamma^\mu_{\delta\gamma}}{\partial x^\mu} - \frac{\partial \Gamma^\mu_{\mu\gamma}}{\partial x^\delta} + \Gamma^\gamma_{\delta\mu} \Gamma^\nu_{\mu\nu} - \Gamma^\mu_{\nu\gamma} \Gamma^\nu_{\mu\delta} \right) \end{aligned} \tag{2.3}$$

Here, the definition of the Christoffel symbols Γ are shown again in Eq.(2.2) to see the dependence of the metric explicitly. Also, the Einstein summation convention must be regarded, which make the equations even more complex. Written out completely, the Einstein equations would consist of a number of terms in order of 10^5. This illustrates the high complexity.

2.1.2 The ADM Equations

The ADM equations make use of the 3+1 decomposition of spacetime. The idea is to decompose the spacetime in an infinite amount of three-dimensional hypersurfaces which each represent space at a different time. Each hypersurface can be labeled with a time coordinate. In the next sections the analytical tools for deriving the ADM equations from the Einstein-Hilbert action are introduced. Then, the constraint equations are further separated in one Hamiltonian constraint equation and three momentum constraint equations. It turns out, that both can be treated independently and the final problem is to solve one semilinear elliptic differential equation. A comprehensive but short introduction gives [Spe01]. An up-to-date summary on the subject of initial data is presented in [Coo91] and finally [MTW73] treats the initial data problem in a separate chapter.

The Foliation of Spacetime

Let us consider a four-dimensional manifold M with coordinates x^α. Then a timelike function $t(x^\alpha)$ parametrizises the collection of hypersurfaces. All $t(x^\alpha) = t_0$, $t_0 \in \mathbb{R}$ refer to one hypersurface which can be labeled with the time t_0. In this way, we gain a foliation of spacetime.

Additional properties must be demanded so that the resulting foliation indeed cover the whole spacetime. We take some special hypersurface and assign a curve $\mathbf{x}(t)$ to each point on this hypersurface. Further we demand that this curves do not intersect each other. Then, we have to demand that the tangent vector of all of these curves are at no point tangent to any of the hypersurfaces to get a valid foliation.

Lapse and Shift

All hypersurfaces of the foliation are best described with a one-form $\mathrm{d}\mathbf{t}$. The tangent vector to this one-form is denoted with δ_t and gives the scalar product $\langle \delta_t, \mathrm{d}\mathbf{t} \rangle = 0$. In Fig.(2.1) it corresponds to the vector $\overrightarrow{PP^+}$. For each hypersurface we define three vector fields $\mathbf{e}_i$ which must be linearly independent at each point. As they represent the basis vectors of the hypersurface at each point they must fulfill $\langle \mathbf{e}_a, \mathrm{d}\mathbf{t} \rangle = 0$. Now, we have constructed a tangent space to each point P which basis is defined by $\{\partial_t, \mathbf{e}_i\}$. How is the metric expressed in this picture of the $3+1$ decomposition? Let us choose an arbitrary vector field $\mathbf{n}$ with $\langle \mathrm{d}t, \mathbf{n} \rangle \neq 0$ and demand further:

(a) $\mathbf{g}(\mathbf{n}, \mathbf{n}) = -1$

(b) $\mathbf{g}(\mathbf{e}_i, \mathbf{n}) = 0$

(c) $\mathbf{g}(\mathbf{e}_i, \mathbf{e}_j) = \gamma_{ij}$

These three conditions guarantee that $\mathbf{n}$ is a timelike unit vector. It emanates perpendicular to the hypersurface and it gives us the 3-metric γ_{ij} for the hypersurface. If we now introduce lapse N and shift vector $\mathbf{N}^i$ defined by

$$\begin{aligned} \partial_t &= N\mathbf{n} + \mathbf{N}^i \mathbf{e}_i \,, \\ \mathbf{n} &= \frac{1}{N}(\delta_t - \mathbf{N}^i \mathbf{e}_i) \,, \end{aligned} \tag{2.4}$$

we can calculate the components of the four-metric $g_{\mu\nu}$:

$$\begin{aligned} g_{00} &= \mathbf{g}(\partial_t, \partial_t) \\ &= \mathbf{g}(N\mathbf{n} + \mathbf{N}^i \mathbf{e}_i, N\mathbf{n} + \mathbf{N}^i \mathbf{e}_i) \\ &= -N^2 + \mathbf{N}^i \mathbf{N}_i \,, \\ g_{0i} &= \mathbf{g}(\partial_t, \mathbf{e}_i) \\ &= \mathbf{g}(N\mathbf{n} + \mathbf{N}^i \mathbf{e}_i, \mathbf{e}_i) \\ &= \mathbf{N}_i \\ g_{ij} &= \gamma_{ij} \,. \end{aligned}$$

This derivation can be summarized shortly in matrix form,

$$g_{\mu\nu} = \begin{pmatrix} g_{00} & g_{0i} \\ g_{i0} & g_{ij} \end{pmatrix} = \begin{pmatrix} -N^2 + \mathbf{N}^i \mathbf{N}_i & 2\mathbf{N}_i \\ & \\ 2\mathbf{N}_i & \gamma_{ij} \end{pmatrix} .$$

or it can be expressed alternatively with the line element

$$\mathrm{d}s^2 = (-N^2 + \mathbf{N}^i \mathbf{N}_i)\mathrm{d}t^2 + 2\mathbf{N}_i \mathrm{d}t\mathrm{d}x^i + \gamma_{ij}\mathrm{d}x^i\mathrm{d}x^j \, .$$

The procedure of introducing lapse and shift is illustrated in Fig.(2.1). The lapse N and its corresponding vector $N\mathbf{n}$ is described with N only in the figure. From the context it is clear if a vector or a parameter has to be considered. From the line element, we see that the minus sign disappears when the shift is chosen too large. In this case ∂_t turns spacelike. Of course, such a case must be omitted with a sufficient small shift vector. Otherwise, the foliation is not valid anymore in the sense of the above formulated criterias. Lapse and shift represent the gauge freedom of the ADM formulation. With a changed lapse and shift, the relations for ∂_t and $\mathbf{n}$ from Eq.(2.4) change only. This yield different curves $\mathbf{x}(t)$. So, lapse and shift can be chosen almost at will.

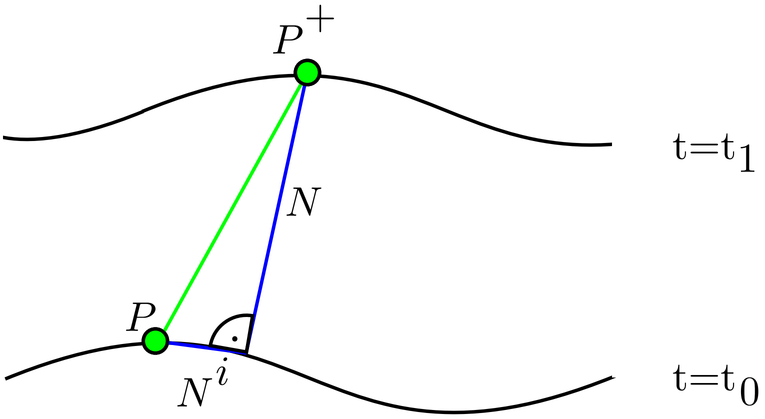


Fig. 2.1: Lapse and shift in ADM formalism. Four degrees of freedom occur when fixing the 4-geometry between two hypersurfaces. Scalar lapse N and 3-shift-vector N^i must be given. Lapse emanates perpendicular to $t = t_0$ hypersurface. $\overrightarrow{PP^+}$ corresponds to the 1-form δ_t

The lapse is the proper time which goes by for an observer who travels with 4-velocity $\mathbf{n}$ from one hypersurface to the next. It can be used to control the advance of each point in time. For instance, we could be interested in slowing down the proper time if an observer corresponding to some spatial point would face a singularity.

The shift vector $\mathbf{N}_i$, which lies within one hypersurface, gives us the projection of the vector ∂_t onto the basis of the hypersurface $\mathbf{e}_i$ by means of the scalar product $\mathbf{g}(\partial_t, \mathbf{e}_i)$.

Projections, Covariant Derivative and Extrinsic Curvature

The $3+1$ decomposition gives us now the possibility to project a tensor onto the hypersurface. The projection of a tensor $\perp T$ is defined as:

$$\begin{aligned} \perp T^{\lambda}{}_{\mu\nu} &= \perp^{\lambda}{}_{\alpha} \perp^{\beta}{}_{\mu} \perp^{\gamma}{}_{\nu} T^{\alpha}{}_{\beta\gamma} \qquad \text{with} \\ \perp^{\mu}{}_{\nu} &= \delta^{\mu}{}_{\nu} + n^{\mu} n_{\nu} \,. \end{aligned}$$

$\delta_{\mu\nu}$ represents the Kronecker symbol. Now, we can see, that γ_{ij} is the 3-metric and it can be calculated by projecting the 4-metric $g_{\mu\nu}$ onto the hypersurface:

$$\begin{aligned} & \gamma_{\mu\nu} = \perp g_{\mu\nu} = g_{\mu\nu} + n_{\mu} n_{\nu} \\ \Rightarrow \quad & \gamma_{ij} = \perp^{\mu}{}_{i} \perp^{\nu}{}_{j} g_{\mu\nu} = g_{ij} \,, \\ & \gamma_{i0} = 0 \,. \end{aligned}$$

In a curved manifold it is more difficult to take the derivative of a function. In contrast to Euclidean space where we have constant basis vectors at each point we have now three vector fields $\mathbf{e}_i$ which in general represents changing basis vectors from point to point. Therefore it is not longer possible to identify the derivative with the derivation of the components itself. Additional terms occur and the change of the basis vectors are represented with the Christoffel symbols. Let ∂_α denote the ordinary partial derivative with respect to the coordinate x^α, then the covariant derivative of a tensor T is defined as

$$\nabla_\delta T^{\alpha\beta}{}_{\gamma} = \partial_\delta T^{\alpha\beta}{}_{\gamma} + \Gamma^{\alpha}_{\rho\delta} T^{\rho\beta}{}_{\gamma} + \Gamma^{\beta}_{\rho\delta} T^{\alpha\rho}{}_{\gamma} - \Gamma^{\rho}_{\gamma\delta} T^{\alpha\beta}{}_{\rho} \,.$$

With the concept of the covariante derivative geodesics can be described by a curve which tangent vector field $\mathbf{v}$ is parallel transported along itself.

$$\mathbf{v}^\alpha \nabla_\alpha \mathbf{v}^\beta = 0 \,.$$

With the covariant derivative and the projection operator the *extrinsic curvature* is defined as

$$K_{\mu\nu} = - \perp \nabla_\mu n_\nu \,.$$

It describes the embedding of the hypersurface into the 4-manifold M. This equation is equivalent to

$$K_{\mu\nu} = -\frac{1}{2} \mathcal{L}_{\mathbf{n}} \gamma_{\mu\nu} \,.$$

The symbol $\mathcal{L}_{\mathbf{n}}$ represents the Lie-derivative along the normal vector field $\mathbf{n}$. The Lie-derivative is another concept of a derivative on a curved manifold. It measures the change of a value with respect to an infinitesimal change of the point at which the value is measured. So, the extrinsic curvature describes how the 3-metric is changing at a given point from one hypersurface to the next. If $K_{\mu\nu}$ does not vanish then an observer moving along $\mathbf{n}$ would measure increasing or decreasing distances between two given points.

The Boundary Value Problem

The variation of the Einstein-Hilbert action can now be realized. The set of variables for the variation is: lapse N, shift N_i and symmetric 3-metric γ_{ij}. There are only six independent components in the symmetric 3-metric. In total, we have 10 variables. The derivation of the constraint problem, which expresses the initial data problem can now be sketched as:

$$\begin{aligned} \frac{\partial S}{\partial N^i} &\overset{!}{=} 0 \quad \Rightarrow \quad \nabla^j (K^{ij} - \gamma_{ij}\mathrm{tr}K) = 0 && \text{momentum constraints}\,, \\ \frac{\partial S}{\partial N} &\overset{!}{=} 0 \quad \Rightarrow \quad R - \mathrm{tr}\ K^2 + (\mathrm{tr}\ K)^2 = 0 && \text{Hamiltonian constraint}\ . \end{aligned}$$

∇_i denotes the spatial covariant derivative compatible to the 3-metric γ_{ij}. tr is the trace and R is the Ricci scalar associated with the 3-metric γ_{ij}. Alternatively, the ADM equations can also be derived from the Einstein equations using projections. Summarizing the right hand side of the Einstein equations with the Einstein tensor $G_{\mu\nu}$,

$$R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} =: G_{\mu\nu}\,,$$

the projection of the Einstein equations onto the hypersurface,

$$\perp G_{\mu\nu} = 0 \quad ,$$

give us the evolution equations. The constraint equations are derived from the two projections:

$$\begin{aligned} G_{\mu\nu} n^\mu n^\nu &= \frac{1}{2}\left(R - \mathrm{tr}\ K^2 + (\mathrm{tr}K)^2\right) = 0 \\ \perp G^{\mu\nu} n_\nu &= -\nabla_\nu K^{\mu\nu} + \nabla^\mu \mathrm{tr}K = 0\,. \end{aligned}$$

Here too, the covariante derivative ∇ and the Ricci scalar R are the values corresponding to the 3-metric γ_{ij}. The constraint equations have the property that they are fulfilled on each hypersurface from the evolution scheme if they are guaranteed on the initial hypersurface. This can be proved with the Bianchi identities.

To find the initial value problem we further follow the York-Lichnerowicz conformal decomposition. First, we choose a static background metric $\tilde{\gamma}_{ij}$ and allow the dynamic 3-metric γ_{ij} to be only this background metric except for a multiplied factor, called the *conformal factor* ψ:

$$\gamma_{ij} = \psi^4\, \tilde{\gamma}_{ij}\,.$$

Now, we can rewrite the Hamiltonian constraint in the vacuum case to

$$\triangle\psi - \frac{1}{8}\psi R - \frac{1}{8}\psi^5 (\mathrm{tr}K)^2 + \frac{1}{8}\psi^5 K_{ij}K^{ij} = 0\,. \tag{2.5}$$

Here, $\triangle = \nabla^i \nabla_i$ is the derivative operator with respect to the background 3-metric $\tilde{\gamma}_{ij}$ and R is the Ricci scalar with respect to the background 3-metric $\tilde{\gamma}_{ij}$. We can further simplify this equation if we make now assumptions concerning the aim to construct initial data for a black hole or two black holes respectively. Far away from the black hole we can assume the metric to be flat. So we can choose the background metric to be flat, also called *conformal flatness*. This brings a lot of simplifications with it. The derivative operator of Eq.(2.5) is now the standard flat Laplacian. Furthermore the term with the Ricci scalar R vanishes, since in flat geometry, $R = 0$.

A second restriction considers a special way of foliating the spacetime. The foliation can be chosen in a way that tr $K = 0$. The extrinsic curvature is demanded to be traceless. This way of foliating is known as *maximal slicing*. Often in literature the symbol A_{ij} is used. It represents the tracefree part of the extrinsic curvature. So, for the case of maximal slicing we can always replace the A_{ij} symbol with K_{ij}.

Considering both assumptions the equations read

$$\triangle\psi + \frac{1}{8}\psi^5 K_{ij}K^{ij} = 0 \tag{2.6}$$

As a last step, the extrinsic curvature K is replaced with its pendant $\tilde{K}$ which measures the extrinsic curvature with respect to the background metric. *The conformal tracefree extrinsic curvature* is defined with the formulas

$$\begin{aligned} K^{ij} &:= \psi^{-10}\tilde{K}^{ij} \quad \text{and} \\ K_{ij} &:= \psi^{-2}\tilde{K}_{ij}\,. \end{aligned}$$

Plugging these definitions into Eq.(2.6) and defining a spatial scalar function $H = \tilde{K}_{ij}\tilde{K}^{ij}$ we get the final equation:

$$\triangle\psi + \frac{1}{8}H\psi^{-7} = 0\,. \tag{2.7}$$

The remaining momentum constraints were solved by J. Bowen and J. York in [BJ80] and they give rise to the function H in Eq.(2.7) when a special problem is considered.

The derived equation is not yet stated correctly because boundary values are not demanded. If we do this and if we further introduce spherical coordinates for the spatial points: $\mathbf{x} = (r, \theta, \phi)$, then for black hole problems as shown in Fig.(2.2) the following problem is obtained:

$$\begin{aligned} \triangle\psi + \frac{1}{8}\cdot\frac{H}{\psi^7} &= 0 \quad \text{for } r \geq a\,, & & \qquad (2.8) \\ \frac{\partial\psi}{\partial r} + \frac{1}{2a}\psi &= 0 \quad \text{for } r = a & & \text{Inner Boundary,} \\ \psi > 0 \ \lim_{r\to\infty}\psi &= 1\,. & & \text{Outer Boundary}\,. \end{aligned}$$

Here, H is a spatial function which specifies the problem to a static, moving or spinning black hole for example. $r = a$ is the position of the event horizon and, from paramount importance, ψ is the conformal factor. Now, it is the only quantity which must be determined. Having calculated this, the problem is solved completely because extrinsic curvature K_{ij} and 3-metric γ_{ij} are determined by means of this factor. As already mentioned, the differential operators in Eq.(2.8) are all operators for flat space.

The initial data for this model applied to a domain in which the black hole is *excised* is also called *Excision Data*. Eq.(2.8) is a semilinear partial differential equation. This means, the

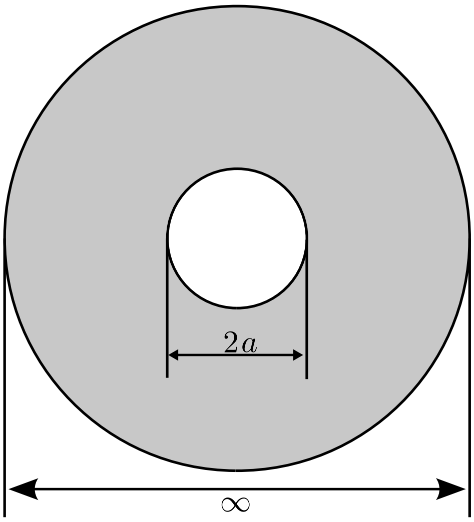


Fig. 2.2: The simplified domain. One spherical coordinate is suppressed. a is the radius of the black hole and $r = a$ is the surface of the inner boundary which conincides with the event horizon. On the outer boundary the metric is flat which is only exactly true at infinity.

equation contains a non-linearity in a factor which does not contain a derivative.

The function $H(\mathbf{x})$ was not determined yet. In the case of a static black hole the term vanishes and the conformal factor can simply be identified from the analytical Schwarzschild metric in isotropical coordinates

$$\mathrm{d}s^2 = -\left(\frac{1 - M/2r}{1 + M/2r}\right)^2 \mathrm{d}t^2 + \left(1 + \frac{M}{2r}\right)^4 (\mathrm{d}r^2 + r^2\,\mathrm{d}\theta^2 + r^2 \sin^2\theta\,\mathrm{d}\phi^2)\,.$$

The term ψ^4 stands right in front of the flat metric expressed here by means of the volume element in spherical coordinates. If we choose $M = 1$, function $H(\mathbf{x})$ and analytical solution $\psi(\mathbf{x})$ is:

$$\begin{aligned} H(\mathbf{x}) &= 0 \\ \psi(\mathbf{x}) &= 1 + \frac{1}{2r}\,. \end{aligned}$$

The conformal factor and the metric component g_{tt} of the Schwarzschild metric is printed in Fig.(2.3). The next two subsections will deal with the function $H(x)$ representing one black hole with linear momentum and a binary black hole problem with linear momentum and angular momentum respectively.

The function $H(x)$ for a Single Black Hole with Linear Momentum

A solution of the momentum constraints provided by Bowen and York in [BJ80] describes a black hole with linear momentum. If we use spherical coordinates (r, θ, ϕ) then a single black hole traveling in z-direction gives rise to

$$H(x) = \frac{9}{2}\frac{P^2}{r^4}\left\{\left(1 \pm \frac{a^2}{r^2}\right)^2 + 2\cos^2\theta\left(1 \pm 4\frac{a^2}{r^2} + \frac{a^4}{r^4}\right)\right\}.$$

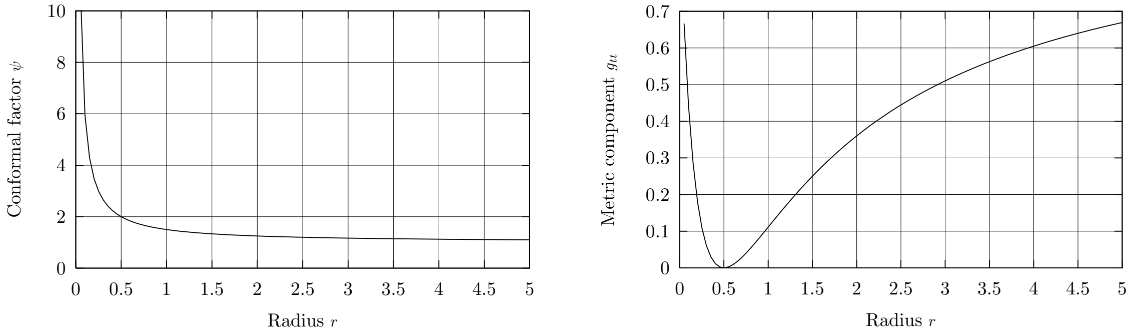


Fig. 2.3: Schwarzschild Solution. Spherical symmetric solution of the conformal factor ψ and the timelike metric component g_{tt} is plotted. *Left.* The conformal factor versus the isotropical radial coordinate r is shown. At the event horizon $r = 1/2$, the conformal factor becomes $\psi = 2$. With higher r the conformal factor converges to the flat value of $\psi = 1$. *Right.* The absolut value of the timelike metric component g_{tt} is plotted versus r. At the event horizon the metric component becomes $g_{tt} = 0$. The speed of convergence to the flat value with respect to r is not as fast as for the conformal factor.

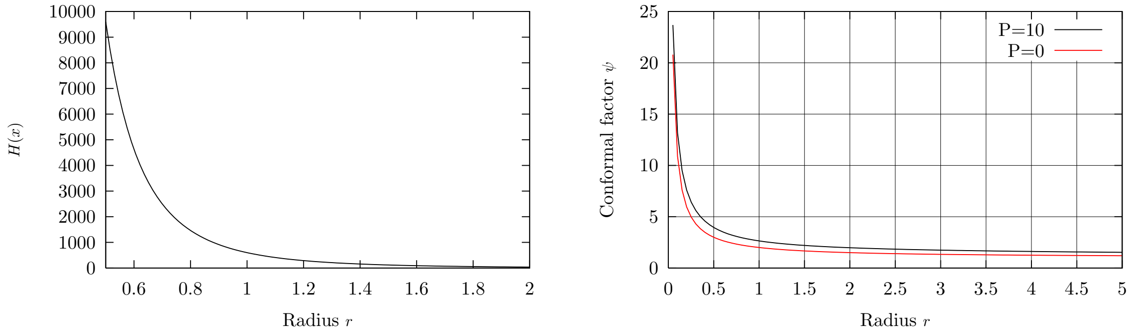


Fig. 2.4: Analytical solution for model problem $P = 10$. *Left.* Function $H(x)$ for a black hole with linear momentum $P = 10$ is shown. The angular dependence is suppressed. *Right.* The solution $\psi(x)$ of the corresponding model problem is shown as well as the conformal factor of the Schwarzschild metric. The conformal factor of a balck hole with linear momentum is always greater than the conformal factor of a static black hole ($P = 0$).

This physical meaningful fuction can be replaced with the so called model problem. It deviates from the physical problem by neglecting the angular dependence. In this way it allows an analytical solving of the differential equation. In this case the function $H(\mathbf{x})$ and the solution $\psi(\mathbf{x})$ is:

$$\begin{aligned} H(r,\theta,\phi) &= 6\frac{P^2}{r^4}\left(1-\frac{a^2}{r^2}\right)^2 , \\ \psi(r,\theta,\phi)^4 &= 1+\frac{2E}{r}+\frac{6a}{r^2}+\frac{2aE}{r^3}+\frac{a^4}{r^4} . \end{aligned}$$

Here, $E^2 = (P^2 + 4a^2)$ is the ADM energy and P represents the linear momentum. a denotes the radius coordinate of the event horizon. In the numerical simulations M is chosen to be equal to one. Fig.(2.4) shows function $H(x)$ and ist solution. Since the problem is not physical anymore, the problem is referred to as *model problem* in [BJ80].

The function $H(x)$ for the Binary Black Hole Problem

In this section a binary black hole problem is considered. For that, the domain shown in Fig.(2.2) must be modified. Given two black holes from different radii a_1 and a_2 and given the separation b of these two black holes, the domain shown in Fig.(2.5) is applied in this thesis. Both black holes are excised from the domain at a radius which coincides with the event horizon of each hole. The boundary conditions for all three boundaries are similar to Eq.(2.8). The outer boundary condition stays the same and the inner boundary condition holds true now for both event horizons. The paramount change comes from the calculation of the function $H(x)$. The derivation is discussed in detail in the paper of COOK ([Coo91]). First we have to decide on a topology representing both black holes since there is no unique generalization for that. Two different approaches are popular. A ususal picture of one black hole is the *Einstein-Rosen bridge* which connects two asymptotical flat universes. Each asymptotical flat universe belongs to positive and negative radii respectively when we use an extended version of the Schwarzschild coordinates. These asymptotical flat universes are also called *sheets*. If we care about N black holes located on one sheet, the corresponding sheets of each black hole can either be the identical sheet for all black holes or each black hole has his own corresponding sheet. In [Coo91] the first method was chosen. This results is a two-sheeted manifold with N throats in general. Further it is assumed, that both sheets are connected with an isometry condition. The value r denotes in the following the radius coordinate if we use spherical coordinates. So, it simply denotes the distance of a point from the origin. Demanding that latin indices run over spatial indices only, COOK derived the following infinite series:

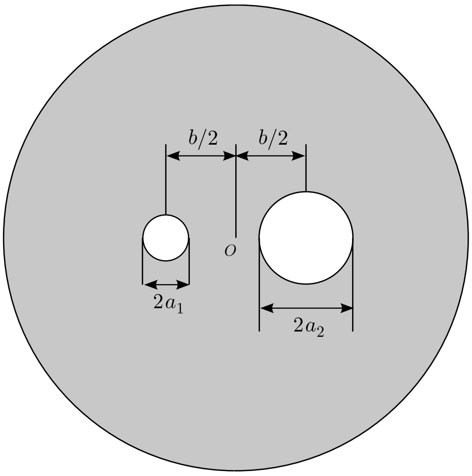


Fig. 2.5: Binary Black Hole Domain. One spherical coordinate is suppressed. a_1 is radius of black hole number 1. a_2 is radius of second black hole, here: $a_2 = 2a_1$. b is the distance between both black holes. O is the origin of the coordinate system. Both black holes are separated along the z-axis. Exact outer boundary is at infinity.

$$\begin{aligned}
\tilde{A}_{ij}(x) &= A_{ij}(x) + \sum_{n=1}^{\infty}(\pm 1)^n \left\{ F_1^n \, (M_1^n)_i^k \, A_{kl} \,|_{(x_1^n)} \, (M_1^n)_j^l + F_2^n \, (M_2^n)_i^k \, A_{kl} \,|_{(x_2^n)} \, (M_2^n)_j^l \right\} \\
{a_1}^4 H(x) &= {a_1}^4 \tilde{A}_{ij} \cdot \tilde{A}^{ij} \qquad (2.9) \\
&= (\tilde{A}_{rr})^2 + (\tilde{A}_{\phi\phi})^2 + (\tilde{A}_{\theta\theta})^2 + 2 \cdot \left\{ \tilde{A}_{rz})^2 + (\tilde{A}_{r\phi})^2 + (\tilde{A}_{\phi z})^2 \right\}
\end{aligned}$$

$\tilde{A}_{ij}$ is called the inversion-symmetric extrinsic curvature. The scalar function F_i^n and the matrices M_i^n and A will be introduced briefly in the following. The upper index n always denotes the summation index. The lower indices 1 and 2 refer to the two different sheets and the indices l and j refer to spatial coordinate indices. If a 2 occurs as an upper index the square has to be taken. If a symbol is follwed by a $|_{(x_1^n)}$ for instance then this symbol is evaluated at the coordinate given by the vector x_1^n.

All symbols with F are called *imaged scale factors.* All symbols M are named *imaged transformation matrices.* First, two maps J_1^i and J_2^i are introduced which identify points in two different sheets. The Jacobian matrices of this map are then defined with

$$\begin{aligned}(J_1)^i_j &:= \frac{\partial}{\partial x^j} J_1^i \\ (J_2)^i_j &:= \frac{\partial}{\partial x^j} J_2^i \,.\end{aligned}$$

The two imaged position vectors $(x_1^n)^i$ and $(x_2^n)^i$ are recursively defined by

$$(x_1^n)^i := \begin{cases} J_1^i \,|_{x_1^{n-1}} & \text{if } n \text{ odd} \\ J_2^i \,|_{x_1^{n-1}} & \text{if } n \text{ even} \\ x^i & \text{if } n = 0 \end{cases}$$

$$(x_2^n)^i := \begin{cases} J_2^i \,|_{x_2^{n-1}} & \text{if } n \text{ odd} \\ J_1^i \,|_{x_2^{n-1}} & \text{if } n \text{ even} \\ x^i & \text{if } n = 0 \end{cases}$$

The symbols $(M_1^n)^i_j$ and $(M_2^n)^i_j$ are recursively defined too with the relations:

$$(M_1^n)^i_j := \begin{cases} (M_1^{n-1})^i_l (J_1)^l_j \,|_{x_1^{n-1}} & \text{if } n \text{ odd} \\ (M_1^{n-1})^i_l (J_2)^l_j \,|_{x_1^{n-1}} & \text{if } n \text{ even} \\ \delta_{ij} & \text{if } n = 0 \end{cases}$$

$$(M_2^n)^i_j := \begin{cases} (M_2^{n-1})^i_l (J_2)^l_j \,|_{x_2^{n-1}} & \text{if } n \text{ odd} \\ (M_2^{n-1})^i_l (J_1)^l_j \,|_{x_2^{n-1}} & \text{if } n \text{ even} \\ \delta_{ij} & \text{if } n = 0 \end{cases}$$

Now, the imaged scale factors F_1^n and F_2^n can be defined with:

$$F_1^n := \begin{cases} (M_1^{n-1})^i_l (J_1)^l_j \,|_{x_1^{n-1}} & \text{if } n \text{ odd} \\ (M_1^{n-1})^i_l (J_2)^l_j \,|_{x_1^{n-1}} & \text{if } n \text{ even} \\ \delta_{ij} & \text{if } n = 0 \end{cases}$$

$$F_2^n := \begin{cases} F_1^{n-1} \left(\dfrac{a_1}{r_1 \,|_{x_1^{n-1}}} \right)^2 & \text{if } n \text{ odd} \\ F_1^{n-1} \left(\dfrac{a_2}{r_2 \,|_{x_1^{n-1}}} \right)^2 & \text{if } n \text{ even} \\ 1 & \text{if } n = 0 \end{cases}$$

In all presented quantities no special coordinate system was employed. Now, we use cylindrical coordinates (ρ, ϕ, z) to characterize the problem further. The black holes are separated along the z-axis and the origin of the coordinate system lies on the mid point of the separation. Then the centers of the black holes C_1 and C_2 are set to:

$$\begin{aligned} \mathbf{C}_1 &= (0 \quad 0 \quad a_1\zeta_1)^T \\ \mathbf{C}_2 &= (0 \quad 0 \quad a_1\zeta_2)^T \end{aligned}$$

So, both centers are measured in units of the radius of the first black hole a_1. Now, the black holes are allowed to have a linear momentum P_1 and P_2 in z-direction as well as to have an angular momentum S_1 and S_2 around the z-axis:

$$\begin{aligned} \mathbf{P}_1 &= (0 \quad 0 \quad P_1)^T \\ \mathbf{P}_2 &= (0 \quad 0 \quad P_2)^T \\ \mathbf{S}_1 &= (0 \quad 0 \quad S_1)^T \\ \mathbf{S}_2 &= (0 \quad 0 \quad S_2)^T \end{aligned}$$

The components of A_{ij} in cylindrical coordinates can be written as:

$$\begin{aligned} A_{\rho\rho} &= -\frac{3}{2} \cdot \left\{ \frac{P_1(z-\zeta_1)^3}{{r_1}^5} + \frac{P_2(z-\zeta_2)^3}{{r_2}^5} \right\} \\ A_{\phi\phi} &= -\frac{3}{2}\rho^2 \cdot \left\{ \frac{P_1(z-\zeta_1)}{{r_1}^3} + \frac{P_2(z-\zeta_2)}{{r_2}^3} \right\} \\ A_{zz} &= \frac{3}{2} \cdot \left\{ \frac{P_1(z-\zeta_1)(\rho^2+2(z-\zeta_1)^2)}{{r_1}^5} + \frac{P_2(z-\zeta_2)(\rho^2+2(z-\zeta_2)^2)}{{r_2}^5} \right\} \\ A_{\rho z} &= \frac{3}{2}\rho \cdot \left\{ \frac{P_1(\rho^2+2(z-\zeta_1)^2)}{{r_1}^5} + \frac{P_2(\rho^2+2(z-\zeta_2)^2)}{{r_2}^5} \right\} \\ A_{\rho\phi} &= \frac{3\rho^3}{2a_1} \cdot \left\{ \frac{S_1}{{r_1}^5} + \frac{S_2}{{r_2}^5} \right\} \\ A_{\phi z} &= \frac{3\rho^2}{2a_1} \cdot \left\{ \frac{S_1(z-\zeta_1)}{{r_1}^5} + \frac{S_2(z-\zeta_2)}{{r_2}^5} \right\} \end{aligned}$$

Here, the following equation for the radius coordinates r_1 and r_2 expressed in cylindrical coordinates hold:

$$\begin{aligned} r_1 &= \sqrt{\rho^2 + (z-\zeta_1)^2} \\ r_2 &= \sqrt{\rho^2 + (z-\zeta_2)^2} \end{aligned}$$

With the abbrevations for ratio of the radii of the black holes and the separation of both measured in units of a_1,

$$\begin{aligned} \alpha &= \frac{a_1}{a_2} \\ \beta &= \frac{|(C_1)^z - (C_2)^z|}{a_1}, \end{aligned}$$

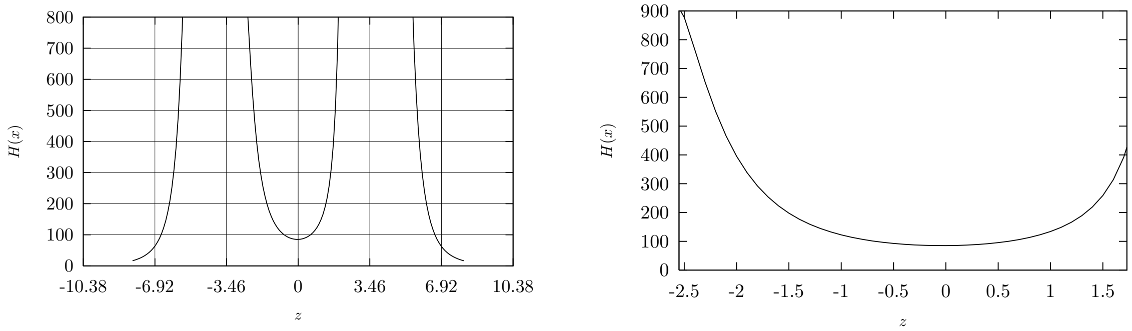


Fig. 2.6: Function $H(x)$ for Binary Black Hole Problem. *Left.* $H(x)$ versus the z-coordinate of cylindrical coordinates. The grid lines are chosen at multiples of the black hole location ($\approx \pm 3.46$). *Right.* $H(x)$ is plotted starting at the event horizon from the first blak hole and ending at the event horizon of the second one.

the isometry maps and the Jacobian of these maps become

$$\mathbf{J}_1 = \begin{pmatrix} \frac{\rho}{{r_1}^2} & \phi & \frac{z-\zeta_1}{{r_1}^2} + \zeta_1 \end{pmatrix}$$

$$\mathbf{J}_2 = \begin{pmatrix} \frac{\rho}{\alpha^2 {r_2}^2} & \phi & \frac{(z-\zeta_2)}{\alpha^2 {r_2}^2} + \zeta_2 \end{pmatrix}$$

$$(J_1)^i_j = \begin{pmatrix} \frac{(z-\zeta_1)^2 - \rho^2}{{r_1}^4} & 0 & \frac{-2\rho(z-\zeta_1)}{{r_1}^4} \\ 0 & 1 & 0 \\ \frac{-2\rho(z-\zeta_1)}{{r_1}^4} & 0 & \frac{-(z-\zeta_1)^2+\rho^2}{{r_1}^4} \end{pmatrix}$$

$$(J_2)^i_j = \begin{pmatrix} \frac{(z-\zeta_1)^2 - \rho^2}{\alpha^2 {r_2}^4} & 0 & \frac{-2\rho(z-\zeta_2)}{\alpha^2 {r_2}^4} \\ 0 & 1 & 0 \\ \frac{-2\rho(z-\zeta_2)}{\alpha^2 {r_2}^4} & 0 & \frac{-(z-\zeta_1)^2+\rho^2}{\alpha^2 {r_2}^4} \end{pmatrix}$$

With these explicit formulas the infinite series from Eq.(2.9) can be solved and can be implemented in a programming language directly. In this thesis, a separate routine `double hamilt(double x)` was written to provide the output value $H(x)$ with any given x. An analytical solution of the conformal factor $\psi(\mathbf{x})$ is now not known anymore. Therefore an exact error can not be derived. This problem is circumvented usually by replacing the exact solution with the solution on the finest grid if we consider convergence issues for example. In Fig.(2.6) the function $H(x)$ is shown along the z-axis. The right hand side shows the function values for points in the outer region between the two black holes.

2.2 The Discretization

The Discretization used in the present work is the well known Finite-Element method. A good introduction into this broad developed field is for example [Joh94] or [GR07]. Furthermore, the FEM for elliptical differential equations is very well known. The next section will shortly sketch the idea of the Finite-Element Method. Afterwards, the already existing code `KASKADE` is presented and modified so it can solve the equation of interest: Eq.(2.8). The use of an developed code is very reasonable. Coding itself would under perfect abilities and conditions yield a code which is only slightly better but would have taken a lot of time to develop.

2.2.1 The Idea of the Finite-Element Method

One of the aims of the Finite-Element Method is to transform the partial differential equation into an algebraic system of equations. This is achieved by formulating ansatz functions. Then, not a whole function is the unknown but the parameters from the ansatz function. This is analogous to replace an infinite set of unknowns (the value of the function at each point of the domain) with a finite set of unknowns (the parameters of the ansatz function).

First, the domain is approximated and decomposed into elements. In this work I will use simplices. The simplicial decomposition of the domain is called *triangulation*. Then, a polynomial ansatz is posed over each element. In this work I will use polynomials of linear order. In general, quadratic or higher degree polynomials are also possible. Additional properties of the solution can be guaranteed with the right ansatz functions. Linear functions over each element give a *piecewise* linear solution over the whole domain. In each element there are four parameters to determine, since the general linear function in three dimensions can be expressed with $f(x, y, z) = a + bx + cy + dz$. The parameters can be calculated from a global linear system of equations which matrix has a band structure. Compared to a system with non-zero elements only, solving band matrices is two orders faster with respect to the number of unknowns.

The following derivation will have a look on the strategy to develop the global linear system of equations from a partial differential equation. Anticipating a linearized formulation of Eq.(2.8) the following model problem is sufficient to look at. It reads

$$\begin{aligned} \triangle u + cu &= f && \text{in } \Omega \subset \mathbb{R}^3, \\ u &= g && \text{on } \partial\Omega_1\,, \\ \frac{\partial u}{\partial n} + pu &= q && \text{on } \partial\Omega_2\,, \end{aligned} \tag{2.10}$$

Here u is the wanted solution, c, f, g, p and q are continuous functions, and $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ with $\partial\Omega_1 \cap \partial\Omega_2 = \{\}$. Now, we multiply with an arbitrary function v, integrate over the whole domain Ω and apply *Integration-By-Parts*. This yield:

$$\int_\Omega (\nabla u \nabla v + c\,uv)\,\mathrm{d}x - \int_{\partial\Omega} \frac{\partial u}{\partial n} v \mathrm{d}s = \int_\Omega f v \mathrm{d}x\,.$$

Considering the boundary conditions of Eq.(2.10) and restricting v to be element of a function space V which functions are zero at the boundary $\partial\Omega_1$ we finally have

$$\int_\Omega (\nabla u \nabla v + c\,uv)\,\mathrm{d}x + \int_{\partial\Omega_2} (pu - q) v \mathrm{d}s = \int_\Omega f v \mathrm{d}x \quad \forall v \in V\,. \tag{2.11}$$

Which other features should the function space V have where the function u and the test functions v come from? In Eq.(2.10) it is differentiated twice at each point, so $u \in C^2(\overline{\Omega})$. Contrary, Eq.(2.11) asks for integrability of the functions u and v and its derivatives. Thats the reason the *Sobolev Space* $H^1(\Omega)$ is introduced. It includes all the functions for which:

$$H^1(\Omega) = \left\{ v \in H^1(\Omega) : ||v||_{H^1(\Omega)} := \int_\Omega ||v(\mathbf{x})||^2 + \sum_{i=1}^{3} \int_\Omega ||\frac{\partial u}{\partial x_i}(\mathbf{x})|| < \infty \right\}.$$

Here, $\mathbf{x} = (x_1, x_2, x_3)^T \in \Omega$. Now, instead of solving Eq.(2.10) to get function u, we can also try to solve the following problem, also known as the *weak formulation* of Eq.(2.10):

$$\begin{aligned}
&\text{Seek } u \in V = H^1(\Omega) \text{ with } u|_{\partial\Omega_1} = g \text{ such that} \\
&\qquad a(u,v) = b(v) \quad \forall\, v \in V \quad \text{and } v|_{\partial\Omega_1} = 0 \qquad \text{with} \\
&\qquad a(u,v) = \int_\Omega (\nabla u \nabla v + c\, uv)\,\mathrm{d}x + \int_{\Omega_2} puv\,\mathrm{d}s\,, \\
&\qquad b(v) = \int_\Omega fv\,\mathrm{d}x + \int_{\Omega_2} qv\,\mathrm{d}s\,.
\end{aligned} \tag{2.12}$$

Here $a(u, v)$ is a bilinear form which is *continuous* and *V-elliptic*. That means, there exists a fixed $M > 0$ and a $\gamma > 0$ so that:

$$\begin{aligned}
|a(u,v)| &\le M||u||\,||v|| \quad \forall u, v \in V\,, \\
a(u,u) &\ge \gamma||u||^2 \quad \forall u \in V\,.
\end{aligned}$$

With this two properties the *Lax-Milgram* lemma proves the uniqueness and existence of Eq.(2.12).

Up to now, no discretization has been introduced. But having Eq.(2.12) the discretization is now straightaway. The infinite dimensional function space $V = H^1(\Omega)$ is replaced with a finite dimensional subspace $V_h(\Omega) \subset V(\Omega)$. This method is known as *Galerkin method.* The structure of V_h depends on the triangulation and the order of the polynomial basis. Choosing linear basis functions the subspace can be described as follows:

$$V_h := \{v \in V_h : v = c_{0i} + c_{1i}x_1 + c_{2i}x_2 + c_{3i}x_3 \quad \forall\,\text{simplices } s_i \in \mathcal{T}\}$$

Here, $\mathcal{T}$ refers to the triangulation and i is the index of the simplex. All c_{ji} are arbitrary constants. They change between different simplices, so the solution is piecewise linear. But it is discontinuous in general. This problem is circumvented by introducing basis functions which guarantee continuousity. In a linear function there are four degrees of freedom per simplex and there are also four nodes per simplex. So it is possible to assign a basis function to each node (instead to each element) and compose the general linear function inside a simplex by means of this basis functions. The whole piecewise linear solution can be expressed now with:

$$\begin{aligned}
v(\mathbf{x}) &= \sum_{i=1}^{N_n} \lambda_i \cdot \phi_i(\mathbf{x})\,, \\
\phi_i(\mathbf{x}_j) &= \delta_{ij} \quad i, j = 1, 2, \ldots, N_n\,.
\end{aligned} \tag{2.13}$$

Here N_n is the number of nodes in the triangulation $\mathcal{T}$ and $\mathbf{x}_j$ denote the location of the node with index j. With the last equation the basis functions are uniquely determined and they guarantee continuousity over the whole domain. How do this functions look like? Vividly, the basis functions are equal to 1 at their corresponding node and then decay linearly to the opposing triangle to 0 in each simplex which hinge at the node. The so defined basis functions are piecewise linear only. They are also referred to as *hat functions* because they look like a

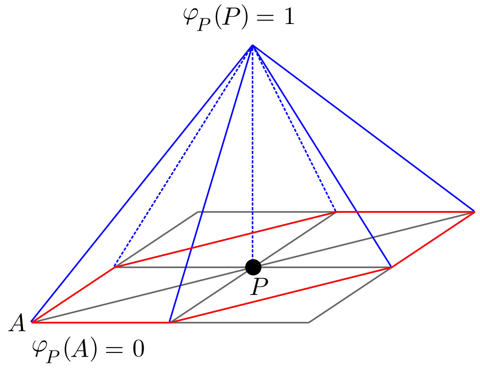


Fig. 2.7: The hat function φ_P corresponding to point P of a 2d triangulation. This linear ansatz for each node guarantees continuousity of the global solution. At point P the function is equal to one. At the red contour and outside the region enclosed by the red contour, the function vanishes. The hat function is piecewise linear.

hat in one- or two-dimensional problems. A two-dimensional example is presented in Fig.(2.7) .

Now, by plugging Eq.(2.13) for functions u and v into Eq.(2.12) the linear system of equations can be derived:

$$\begin{aligned}
& & a(u,v) &= b(v) & \forall v \in V_h \\
&\Rightarrow & a\left(\sum_i \lambda_i\,\phi_i, \sum_j \lambda_j\,\phi_j\right) &= b\left(\sum_j \lambda_j \phi_j\right) & \forall \lambda_j \in \mathbb{R} \\
&\Rightarrow & \sum_j \lambda_j \sum_i \lambda_i\, a(\phi_i,\phi_j) &= \sum_j \lambda_j\, b(\phi_j) & \forall \lambda_j \in \mathbb{R} \\
&\Rightarrow & \sum_i \lambda_i\, a(\phi_i,\phi_j) &= b(\phi_j) & \\
&\Rightarrow & A \cdot \boldsymbol{\lambda} &= \mathbf{b} &
\end{aligned}$$

In the last equation A is a quadratic N_n-matrix with $a(\phi_i,\phi_j)$ as components, $\boldsymbol{\lambda}$ collects the λ_i and $\mathbf{b}$ collects the $b(\phi_j)$ in a vector of same dimension. In general, this linear system is huge, solving for all coefficients simultaneously. Its sparse structure allows efficient methods for solving like multilevel methods. Direct solvers would need $\mathcal{O}(N^3)$ effort to solve a system of equations with a full matrix. A system with a sparse matrix needs only $\mathcal{O}(N)$ effort to be solved. This is a huge difference.

2.2.2 The Program KASKADE and its Modifications to Solve the Initial Data Problem

`KASKADE 3.2.1` was developed from Rudolf Beck, Bodo Erdmann and Rainer Roitzsch (see [BER95]) for the solution of linear partial differential equations in one, two and three dimensions. It features an object-oriented implementation concept employing the programming language `C++`. It works highly reliable and implements different solvers and preconditioners. It makes use of linear and quadratic basis functions and takes simplex triangulations exclusively. In the present work only linear basis functions are considered.

To solve the initial data problem from Eq.(2.8) it is necessary to first linearize the problem. Given the right start values, a fix point iteration solves for the solution of the semilinear problem. Existence and uniqueness of a solution for the corresponding class of semilinear problems achieved with this iteration scheme are proved in [Pao92]. It turns out, that the iteration scheme succeeds and converges very fast in practice.

The main effort to implement this iteration in the `KASKADE` code is the need to understand this highly structured code and its mode of operation. Whether the code is very good formatted and maintained, object-orientedness, complex composed data types, a lot of pointers and a rudimentary documentation make this task not easy for a non-professional programmer.

Let us now derive the linearized version of Eq.(2.8) for the iteration scheme. The semilinear term ψ^{-7} is linearized with a Taylor expansion. Assuming a start solution $\tilde{\psi}$ a linear expansion yields

$$\begin{aligned} & \psi^{-7} = \tilde{\psi}^{-7} + \frac{\partial \psi^{-7}}{\partial \psi}[\psi = \tilde{\psi}] \cdot (\psi - \tilde{\psi}) + \mathcal{O}(\psi - \tilde{\psi})^2 \,. \\ \Rightarrow \quad & \psi^{-7} \approx 8\tilde{\psi}^{-7} - 7\tilde{\psi}^{-8}\psi \,. \end{aligned}$$

Putting this result into Eq.(2.8), we get the following linear problem, which solution represents one iteration step. The outer boundary conditions for infinite radius were replaced with approximated boundary conditions at a finite radius:

$$\begin{aligned} \triangle\psi - \frac{7H}{8\tilde{\psi}^8}\psi &= -\frac{8H}{\tilde{\psi}^7} && \text{for } R \geq r \geq a \,, \\ \frac{\partial\psi}{\partial r} + \frac{1}{2a}\psi &= 0 && \text{for } r = a \qquad \text{Inner Boundary,} \\ \frac{\partial\psi}{\partial r} + \frac{1}{r}\psi &= \frac{1}{r} && \text{for } r = R \qquad \text{Outer Boundary}\,. \end{aligned} \tag{2.14}$$

H and $\tilde{\psi}$ are both fixed coefficient functions. The algorithm 1 shows the iteration scheme applied to solve the semilinear initial data problem.

Algorithm 1 Iteration Scheme to Solve Semilinear Equation with Linear Solver.

```
1: Give start solution $\tilde{\psi}$
2: Calculate Error
3: while Error < TOL do
4:     Solve linear problem in $\psi$ with Eq.(2.14)
5:     Calculate Error
6:     Update start solution $\tilde{\psi} \leftarrow \psi$
7: end while
```

Line `3` solves the linear problem as long as an appropriate error measure falls below the tolerance `TOL`. For the error measure the L_∞ norm with respect to the finite dimensional V_h was applied:

$$\text{Error} := \max_{i=1,\ldots,N_{\text{Nodes}}} ||\psi(\mathbf{x}_i) - \tilde{\psi}(\mathbf{x}_i)|| \,.$$

The iteration was now implemented into the `KASKADE` code. Normally, this code can solve, among others, problems of the following kind:

$$\begin{aligned} -\nabla k \nabla u + qu &= f && \text{in } \Omega \\ u &= u_0 && \text{on } \Gamma_D \\ k\frac{\partial u}{\partial n} &= q_N && \text{on} \Gamma_N \\ k\frac{\partial u}{\partial n} + \alpha u &= q_c && \text{on } \Gamma_C \end{aligned}$$

k, q, α and f are parameters and may depend on spatial coordinates. Ω is the region on which the differential equation is supposed to be solved. Γ_D and Γ_N describe the boundaries of Dirichlet or Neumann type respectively. And Γ_C represents a boundary on which Cauchy type boundary conditions are posed. The latter case is used in the black hole problem from Eq.(2.14) twice: Once at the inner boundary and second at the outer boundary too.

Regarding the semilinear equation Eq.(2.14) we can set $k = 1$ and choose the parameters q, α and f. The terminology gives us the description *mass term* for the function $q(x)$ and gives us the name *source term* for the function $f(x)$. These names are used often in the `KASKADE` code, so it is useful to know them. Finally, we are ready to start the implementation of the iteration scheme according to algorithm 1.

In its original version the program does not allow an easy implementation of an iteration scheme. A lot of procedures have to be changed so that the code is working properly. Another problem was posed by the GCC compiler used to translate sources into code executable by the machine. Despite its great development some routines are not allowed anymore when using `C++`. For instance inherited templates are forbidden in the actual version of the compiler. So, for this thesis changes in high abstraction and also changes on a low level were necessary if memory issues account for that. To solve the black hole problem with `KASKADE` the following files of the package were changed:

- `stack.c` , `alloc.c`
  Low-level changes of the memory routines were necessary at this point to make the compilation with an up-to-date GCC compiler possible.

- `elements.c` , `elementsA.c`
  The class routines `assembleMassSL` and `assembleSourceSL` were included to assemble the right terms within the iteration scheme to solve the semilear equation. Both methods are applied directly in `problemstat.cc`. `assembleMassExact` was changed to calculate the function $H(x)$. It need to be calculated at this place as well as in `userStatic.cc`. This is due to the object structure of `KASKADE`, where no access to routines from `userStatic.cc` are possible at this place.

- `intB.cc`
  The multigrid refinement class `MGInt` had to be changed since otherwise the grid is refined automatically after one iteration step for the used iteration scheme to solve semilinear equations.

- `kaskade.cc`, `linsystem.cc`, `linsystemA.cc`, `problem3.cc`
  Small changes like an adjustment in the calculation of the norm of the solution were done.

- `problem.cc`
  The main changes were applied to this source code file. It treats the iteration loop for the solution of seminlinear equations. It calculates minimum and maximum angles and write data output to the hard disk. The algorithm 1 was applied here.

- `problemstat.cc`
  Here, the class routines `elem.assembleMass` was replaced with `elem.assembleMassSL` and the routine `elem.assembleSource` was replaced with `elem.assembleSourceSL`

- `triang3.cc`
  When the Red-Green refinement technique (described shortly in section 2.4) is applied to the triangulation, new points are generated and these new points may reside at one of the two boundaries used in the excised black hole domain. Such points are projected onto spheres of the appropriate radius in this routine.

- `userStatic.cc`
  In this file the source or mass term respectively can be included easily. This point is described in detail in the user manual of `KASKADE`. The routines `UserStaticMaterial:: M3d` is used to include the mass term and the routine `UserStaticMaterial:: S3d` is used to include the source term. The boundary conditions were implemented with the routine `UserStaticMaterial:: Cauchy`.

To make variables accessible at different points in the developed class hierachy took a significant amount of effort. For instance, the assembling of the mass and source term in `elementsA.cc` depends on the spatial coordinates. These must be made avaiable in all routines which use them.

For the calculation of binary black hole domains only a small amount of additional changes were needed. The routine `triang3.cc` was changed to project onto three occuring boundaries. Also the file `userStatic.cc` was changed to implement three different Cauchy boundaries. The main effort took the implementation of the function $H(x)$ from Eq.(2.14). A separate library was written which was then included into the `KASKADE` code.

2.3 Simplicial Decompositions of The Three-Dimensional Domain

When restricting finite elements to simplices and restricting the polynomial basis to a linear basis, there is still a lot of freedom, namely in the choice of the triangulation with simplices. Triangulations of two dimensional domains can be managed directly since the result is visually directly accessible. But already in three dimensions the task of triangulation turns out to be complicated or at least a little bit unclear at first glance. It would be easier to think of filling a three-dimensional domain with cubes instead of filling this domain with tetrahedrons. This approach leads to the problem of triangulating a cube and is discussed in the next section. Combining the different presented techniques I found a flexible triangulation for the initial data domain from Fig.(2.2). This triangulation will be called `MESHGEN` in the present work. It will be presented in the second section. The last section will deal with another approach: The use of external mesh generators. The effort of constructing grids for quite complicated domains like the black hole domain or binary black hole domain is transfered to an external developed program. Here, `NETGEN` is applied.

A triangulation $\mathcal{T}$ of a three-dimensional domain Ω will consist of 0-dimensional simplices (points), 1-dimensional simplices (edges), 2-dimensional simplices (triangles) and 3-dimensional simplices (tetrahedrons). The set of all i-dimensional simplices will be noted with Σ_i. It is important that triangulations are *valid*. In the present work, the term *valid triangulation* describes the following aspect:

$$\begin{aligned}&\text{A triangulation is called \textit{valid triangulation}, if:}\\ &\qquad p \in \Sigma_i \text{ and } q \in \Sigma_i \quad \Longrightarrow \quad (p \cap q) \in \Sigma_j \text{ with } j < i\,, \quad i = 1,2,3\,.\end{aligned} \tag{2.15}$$

2.3.1 Triangulations of the 3-Cube

There are several methods to triangulate a three-dimensional cube. Three methods will be discussed: First, Kuhn's triangulation, second, the prism method and third, a method to triangulate the 3-cube with the smallest possible number of simplices. The first two methods will need six simplices to triangulate a cube. The last one will need only five simplices.

Kuhn's method is one of the easiest ways to triangulate a cube. Let us assume a unit cube $[0,1]^n$ of arbitrary dimension n. In general, each simplex has the point $A = (0,0,\cdots,0)^T$ and $Z = (1,1,\cdots,1)^T$ in common. The other points of each simplex can be represented with a permutation of the n vector indices. If we restrict Kuhn's method to three dimensions each simplex has the points $A = (000)$ and $Z = (111)$ in common. The 3-permutations $(p_1 p_2 p_3)$ give us the remaining two points. Let x_i denote the i-th component of a vector $\mathbf{x}$, then one remaining point has coordinate zero everwhere, except for $x_{p_1} = 1$. The last point has coordinates zero everywhere, except for $x_{p_1} = x_{p_2} = 1$. For example, the permutation 132 would give the simplex $\{A, Z, (100)^T, (101)^T\}$. The number of permutations in three dimensions is $3! = 6$, and in n dimensions the number is $n!$. The resulting triangulation is unique. It is shown in Fig.(2.8) and summarized in Tab.(2.1).

Permutation	2nd point	3rd point
123	$(100)^T$	$(110)^T$
132	$(100)^T$	$(101)^T$
213	$(010)^T$	$(110)^T$
231	$(010)^T$	$(011)^T$
312	$(001)^T$	$(101)^T$
321	$(001)^T$	$(011)^T$

Tab. 2.1: The six simplices from Kuhn's method in three dimensions excluding the common points A and Z.

Let us now introduce the *prism method*. In three dimensions it starts with a triangulation of a square (2-cube). A triangulation of a square is trivial. The only question is about the orientation of the diagonal which can switch between two positions. Given the square $A = (0,0), B = (0,1), C = (1,0), D = (1,1)$, the triangle sets $\{ABC, BCD\}$ and $\{ACD, ABD\}$ triangulate the square in a valid way. Given one set of triangles, a prism is constructed in z-direction over each triangle. Each prism can be constructed independently from each other but for a valid triangulation the faces which connect two prisms must be triangulated in the same way. Otherwise, the triangulation would be not valid. This problem vanishes, when a modified scheme is applied. Let $\mathcal{T}$ denote the triangulation of the 3-cube and $\mathcal{T}_{-}$ denote the triangulation of the square or 2-cube respectively. Let A, B, C, D denote the points at $z = 0$

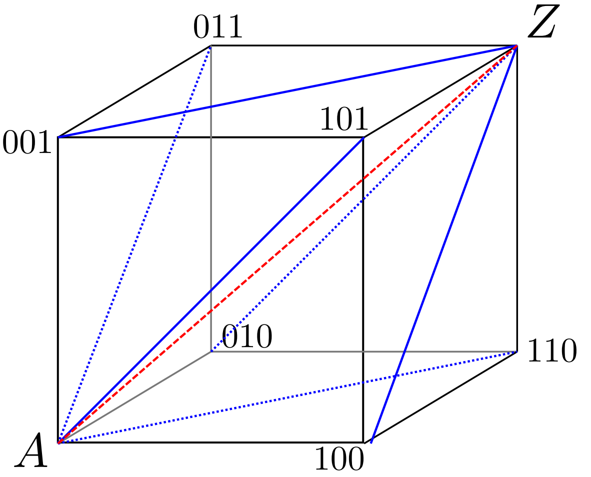


Fig. 2.8: A 3-cube triangulated with Kuhn's method..

and let A^+, B^+, C^+ and D^+ denote the points of the 3-cube at $z = 1$. Then the algorithm 2 shows the general concept. The prism method is applicable in any dimension. Since it is worth mentioning here, but it is used in the next chapter only, details and an example is postponed to section 3.2.2, as well as an illustration in Fig.(3.9).

Algorithm 2 Modified prism method restricted to three dimensions.

```
Give triangulation $\mathcal{T}_-$ of square
Triangulation of $z = 0$ according to $\mathcal{T}_-$
Triangulation of $z = 1$ according to $\mathcal{T}_-$ {Now, the upper and lower face of the 3-cube is
  triangulated already and all points have been introduced.}
$\mathbb{M} = \{\}$
$\mathcal{T} = \{\}$
for all Points $p \in \mathcal{T}_-$ do
  for all Triangles $T$ with $p \in T$ do
    $S$ = Points of $T$ { $S$ denote set of points of new simplex }
    for all Points $q$ of $S$ do
      if $q^+ \in \mathbb{M}$ then
        Replace $q$ in $S$ with point $q^+$
      end if
    end for
    $S = S + p^+$
    $\mathbb{M} = \mathbb{M} + p^+$
    $\mathcal{T} = \mathcal{T} + S$
  end for
end for
```

The last method concerns the aim of constructing a triangulation of the 3-cube with the smallest number of tetrahedrons possible. The idea behind can also be applied in four dimensions to minimize the number of 4-simplices needed to triangulate a 4-cube. It is about *chipping-off* corners. In three dimensions it can be easily shown with Fig.(2.9). Here, the corners A, D, F and G were *chipped off*, giving the four simplices $ABCE, BDCH, BEFH$ and $CEHG$. These four tetrahedrons completely triangulate all six faces of the 3-cube. The fifth tetrahedron $CBEH$ do not have any triangle with the boundary of the cube in common but it triangulates a part of the inner of the 3-cube only. Anticipating the application of chipping-off corners in higher dimensions, it makes sense to distinguish between *outer simplices* which have faces with the boundary of the cube in common and *inner simplices* which naturally consist of diagonals only (2-diagonals, 3-diagonals, ...). The first both triangulations, namely Kuhn's method and the prism method yield triangulations which consist of outer simplices only.

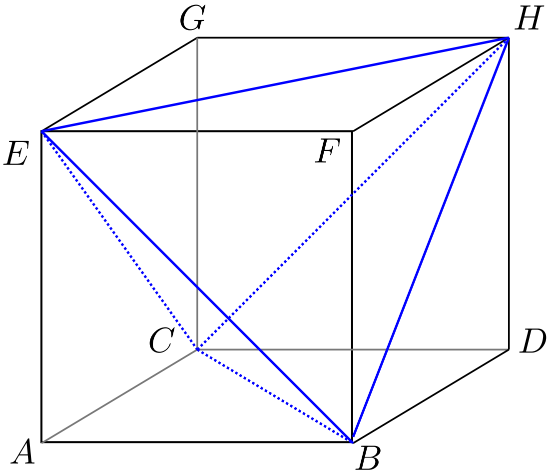


Fig. 2.9: A triangulation of the 3-cube which needs only five simplices.

2.3.2 The Structured Triangulation MESHGEN

The aim of this section is to construct a triangulation for the excision topology shown in Fig.(2.2) with finite outer radius by means of the idea of triangulating cubes. The triangulation is found in a natural way. First, a ball is triangulated by triangulating $n \times n \times n$ cubes and projecting the boundary points onto the boundary of the ball. Especially for $n = 1$ the points are already projected onto a sphere.

Unfortunately the domain of interest consists of a huge ball B_R minus a very small ball B_a. At first glance, this can be modelled easily by leaving away the cube in the middle. But this naive approach poses two problems:

(1) The maximum cube edge length is restricted to $2a$. Because the later employed model is supposed to guarantee $R \gg a$, the number of cubes needed to cover the whole domain is dramatically high. In this way, no use of the flexibility of the FEM is made.

(2) The projection of points onto the boundary of the huge ball B_R is no problem because the outer boundary is convex. Contrary, projections of points onto the concave inner boundary, that is the boundary of the excised ball B_a, poses a problem.

The first point can be circumvented completely with technical effort. The idea for that is sketched in Fig.(2.10) and Fig.(2.11). The method will be described now by means of a fixed grid resolution but the method can be applied directly with higher grid resolutions as well. Here, a $3 \times 3 \times 3$ grid is used, which consists of 27 cubes. They constitute a so called *supercube*. The aim is to scale the cube sizes, beginning with small cubes near to the inner boundary and ending with large cubes at the outer boundary. With this the number of cubes or the number of simplices respectively reduces dramatically compared to the problem stated in point one from the list above. First, the cube in the middle is erased to model the excised ball B_a. Then, all cubes on diagonals are erased, if we take the origin of both balls as the origin of the coordinate system. This gives us six cubes. The resulting triangulation should fill again the supercube completely with simplices. So, opposite points are identified and are moved to the cube edges of the supercube. Because points have been identified, new faces from the different cubes touch each other. They must be triangulated in the same way because the triangulation must be valid. For that, each cube is triangulated with Kuhn's method and then the cube is rotated around the three axis in an appropriate way.

Overcoming this technical problem, the scheme is applied for many levels (5 to 10). In this way, the edge lengths of the cubes get three times larger in radius coordinate direction. So, near to the outer boundary the cubes are very large and the cubes near to the inner boundary are small. This respects the expected solution because on the outer boundary the metric is expected to be flat ($\psi \approx 1$). Then, the metric deviates from flatness more and more until the inner boundary is reached ($\psi \approx 2$).

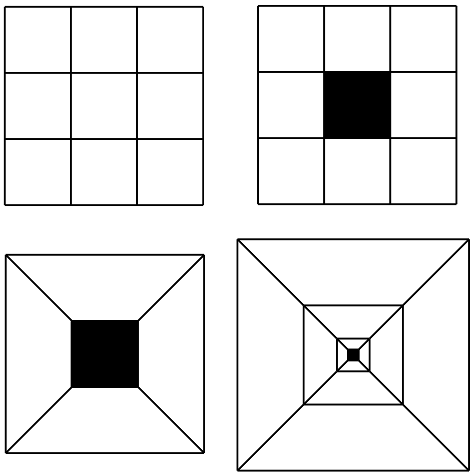


Fig. 2.10: MESHGEN net in a two dimensional abstraction. *Top Left.* 3x3 cubes compose a supercube. *Top Right.* Middle Cube is left away to model inner boundary. *Bottom Left.* Diagonal cubes are left away, points are identified and are moved to the supercube edges to cover the excised supercube completely with simplices. *Bottom Right.* Applied successively, the triangulation cascades from inner boundary to far away outer boundary.

The second point, the projection of points onto a concave boundary, can not be solved completely. This is because the code `KASKADE` uses grid refinement techniques (e.g. it decompose one big tetrahedron in many smaller). This is standard and very powerful since it speeds up the code dramatically. But for a successful refinement a reasonable fine initial triangulation near to the concave boundary is necessary. The issue of too coarse initial triangulations is illustrated in Fig.(2.12). Even if a sufficiently fine triangulation at the concave boundary is guaranteed, maximum and minimum angles are not bounded anymore. In detail this means, with each additional refinement step the maximum angle gets closer to π and the minimum angle gets closer to 0.

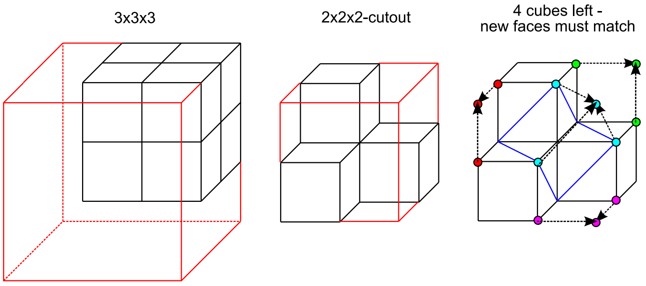


Fig. 2.11: MESHGEN net. From a 3x3x3 supercube triangulation a 2x2x2 cutout is focused at. Cubes along 3-diagonals are erased and special points are identified and moved to the cube vertices of the supercube (arrows pointing to vertices of supercube). All points of same colour represent one identical point after identification. By identifying points new two-dimensional faces touch each other. These three faces (shown with blue diagonals) must be triangulated in the same way. Rotated versions of Kuhn's triangulation must be applied to guarantee correctly triangulated cube faces. This is fulfilled here.

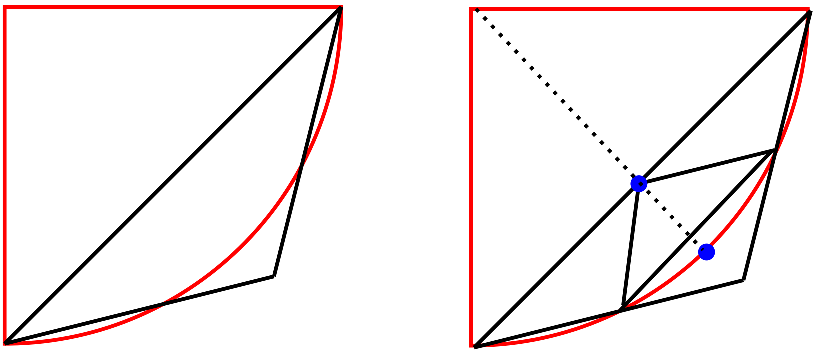


Fig. 2.12: Refining concave boundaries together with radial projection can crash. *Left.* Too coarse initial triangulation. *Right.* Refined triangle (black) introduces one additional boundary point (blue). If the point is projected onto the boundary (red) than the triangulation becomes invalid.

2.3.3 The Unstructured Triangulation from the External Mesh Generator NETGEN

To model manually a structured grid for the single black hole problem shown in Fig.(2.2) is already a task with significant effort. In principle it is also possible to model a binary black hole system with methods used for generating the `MESHGEN` mesh. But it is much faster to employ an already developed grid generator for finite elements. To my best knowledge, there exists no free standardized program which is able to deal with three-dimensional grid generation and covers all methods known. So, a special program must be applied. In the present work, I make use of the `NETGEN` grid generator. It is administered by Joachim SCHOBERL and is freely available under the GNU Library or Lesser General Public License (LGPL)[1]. The generated grid is completely unstructured and optimized with respect to a mostly good angle condition (e.g. to minimize the maximum angle and to maximize the minimum angle).

---

[1] Available at sourceforge.net

2.4 Special Problems and Results

In this section I will show and discuss the results from the modified `KASKADE` code applied to problems in Numerical Relativity concerning initial data. The section will focus on initial data for black hole constellations in three cases: First, a single static black hole, second, one black hole with linear momentum and third, a binary black hole system for head-on collision is considered. The results of the simulations depend mainly on two parameters: Grid refinement strategy and initial triangulation. The latter gets the attribute initial since the triangulation is refined during calculation. Both parameters will be described now in detail.

Let us first consider the grid refinement strategy. For grid refinement an *uniform* and an *adaptive* refinement strategy was chosen. The uniform refinement method refines every element according to a refinement scheme. The adaptive refinement detects regions of the domain which needs to be refined automatically by means of an error estimator. Naturally, the number of simplices will raise slower in the adaptive scheme than in the uniform scheme. `KASKADE` employes an error estimator which is described in [Lei90]. In theory, there exist mainly two different refinement schemes from which only the latter one will be used in this work: Bisection and Red-Green refinement.

The main idea of bisection is to make two elements from one. If an element is chosen to be refined usually the longest edge is selected and halved, introducing one new point. New simplices corresponding to this point are added to the triangulation. This method is described for example in [Mit89] and for higher dimensions in [Mau95]. It will not be used in the present work.

The main idea of Red-Green refinement is described in [Mit89]. Here, each edge which is chosen to be refined is halved. In this way, new points are introduced and new simplices are built upon. `KASKADE` makes use of this idea. The refinement technique is described in detail in [Lei90]. If we consider the simplices of the triangulation and refine every edge of a simplex, then in two dimensions, a triangle is divided into four triangles. In three dimensions, one tetrahedron is refined to eight new ones. So, the raise of elements in one refinement step of Red-Green refinement depends on the dimension. It is much higher than in the bisection scheme where in any dimension one coarse simplex yield only two refined simplices.

Having halved the selected edges and having added the corresponding geometrical objects, the refined triangulation is not valid in general. In a second step additional not-selected edges are halved to make the triangulation valid again. This is called the *closure procedure*. It is more complicated in the bisection scheme than in the Red-Green refinement scheme. Here, I do not go into detail any further.

Let us now consider the initial triangulation. In the present work two different grids were used for that. First, the one introduced above with a structured approach called `MESHGEN`. Second, the grid generated with the external grid generator `NETGEN`. In Fig.(2.13) plane cuts through the three-dimensional initial triangulations are shown in a special case.

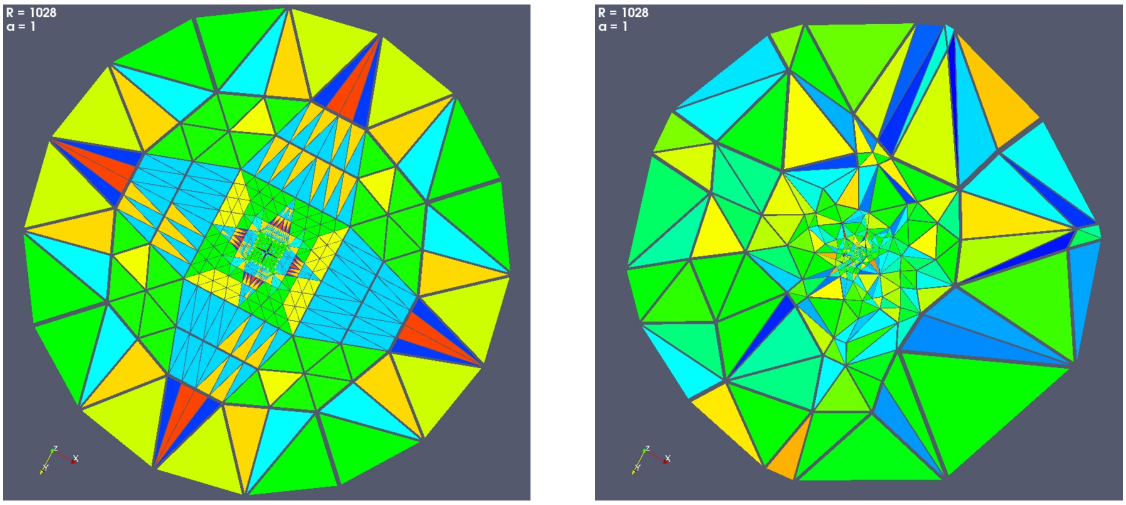


Fig. 2.13: Single Black Hole Problem. Event horizon at $r = a$. Inner Radius = $1a$. Outer Radius = $1024a$. *Left.* A plane cut through the MESHGEN grid. *Right.* A plane cut through the NETGEN grid. Color indicates small angles (all good despite red color).

2.4.1 Single Black Holes

The easiest case to consider is a non-rotating and non-moving black hole. Its analytical solution was found by Karl SCHWARZSCHILD in 1916. In isotropical coordinates its line element reads

$$\mathrm{d}s^2 = -\left(\frac{1 - M/2r}{1 + M/2r}\right) \mathrm{d}t^2 + \left(1 + \frac{M}{2r}\right)^4 (\mathrm{d}r^2 + r^2\,\mathrm{d}\theta^2 + r^2 sin^2\theta\,\mathrm{d}\phi^2)\,. \tag{2.16}$$

Here, M is the ADM mass. From the metric component in front of $\mathrm{d}t^2$ it can be seen that it vanishes when $r = M/2$. This sphere is called the event horizon. The spatial part is a factor multiplied with the flat spatial metric in spherical coordinates. This factor is a power of the conformal factor: ψ^4. The function $H(\mathbf{x})$ in the initial data problem from Eq.(2.8) vanishes in the case of a static black hole. The resulting problem is a simple Laplace problem:

$$\begin{aligned} \triangle\psi &= 0 \quad \text{for } R \geq r \geq a\,, && \\ \frac{\partial\psi}{\partial r} + \frac{1}{2a}\psi &= 0 \quad \text{for } r = a && \text{Inner Boundary,} \\ \frac{\partial\psi}{\partial r} + \frac{1}{r}\psi &= \frac{1}{r} \quad \text{for } r = R && \text{Outer Boundary}\,. \end{aligned}$$

This can be solved directly with the code `KASKADE`.

The function $H(\mathbf{x})$ does not vanish if the black hole travels with linear momentum. The function of the model problem which suppresses terms of the correct $H(x)$ for sake of an analytical soultion is derived in section 2.1.2. In Fig.(2.14) a plane cut through the spherical symmetric solution is plotted. Seven uniform refinement steps were calculated to get this result. It encodes with color the conformal factor ψ. For the static black hole, ψ raises from about 1 on the outer sphere to 2 on the inner sphere at the event horizon. This can be expected from the analytical solution. In the case of a black hole with linear momentum, $P = 10$ was chosen and the conformal factor increases compared with the static case.

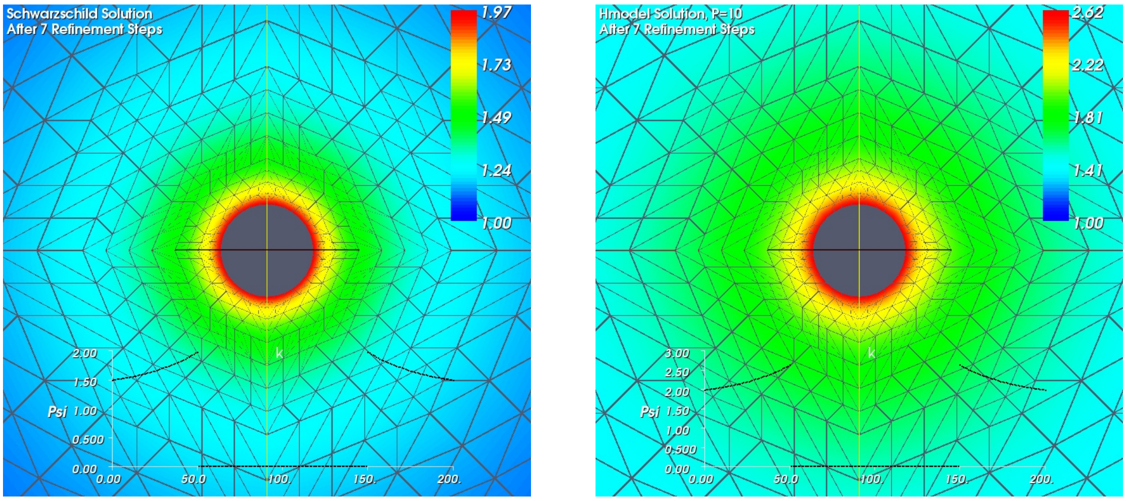


Fig. 2.14: Single Black Hole Problem. Event horizon at $r = a$. Inner Radius $= 1a$. Outer Radius $= 1024a$. *Left.* A plane cut through solution with $P = 0$ (MESHGEN/uniform). *Right.* A plane cut through the solution with $P = 10$ (MESHGEN/uniform).

The semilinear problem $P = 10$ gives a good occasion to compare the grids `MESHGEN` and `NETGEN`. It can be expected that `MESHGEN` is in favor because the grid respects the symmetry better. The left plot in Fig.(2.15) shows that the expectation is fulfilled but only to a small extent. In the figure, h, usually a measure of the edge length in a triangulation, is simply chosen to be $h = 1/N$ with the number of nodes N in the triangulation. It represents also computational cost because this raises proportional to $N \log N$ (`KASKADE` is developed in a way that guarantees an optimal performance). The y-axis in the figure represents the error to the exact solution in the *energy norm*:

$$\text{Error} := a(u_h - u_{\text{exact}}, u_h - u_{\text{exact}})\,. \tag{2.17}$$

a denotes the bilinear form of the weak formulation, u_h is the numerical solution and u_{exact} is the analytical solution.

On the one hand, the result means that the manual generation of a grid to solve the initial data problem is a success. On the other hand, it also states, that a manual grid generation is not forced, since the convergence ratio as well as the magnitude of error of both approaches is similar. This is a reasonable argument to drop manual grid generation for more complex cases like the binary black hole problem.

The next point investigated considers the difference between uniform grid refinement and adaptive grid refinement. It reveals which method gives a better result with respect to the energy error (Eq.(2.17)). On the right hand side of Fig.(2.15) the result is shown. In this special case there is almost no difference between the adaptive and the uniform approach. Whether the energy error of the adaptive solution is always greater than the error of the uniform solution a different advantage of the adaptive refinement can be seen. Since the number of unknowns gets higher only slightly within a single adaptive refinement step it can be assumed that adaptive refinement falls below a given error first. Because in contrast, the number of unknowns with uniform refinement increases by a factor of eight in each refinement step.

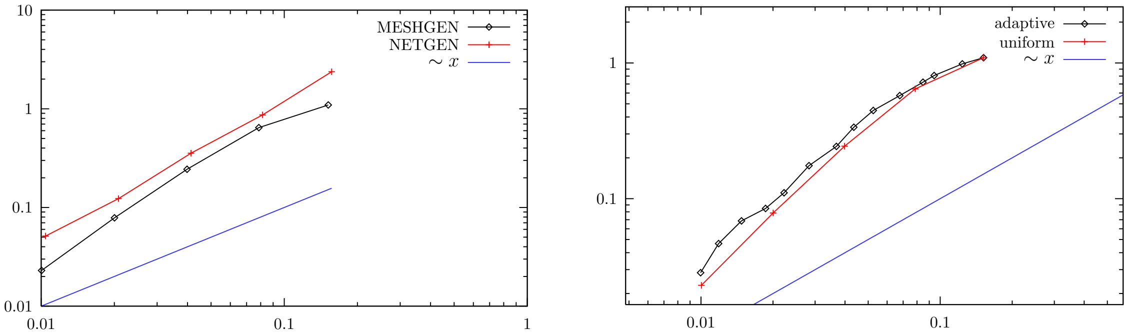


Fig. 2.15: Errors in Single Black Hole Problem. $P = 10$. x-axis: $h = 1/N_{\text{nodes}}$. y-axis: energy error. *Left.* MESHGEN vs. NETGEN (uniform). *Right.* Uniform vs. Adaptive mesh refinement (MESHGEN).

2.4.2 A Binary System of Black Holes with Similar Masses

In this section the solution to a binary black hole problem with almost equal masses are presented. The topology of the domain changes. Now, the huge ball B_R has two excised smaller balls B_{a1} and B_{a2}. The boundary conditions does not change a lot. The outer boundary condition is identical to Eq.(2.14) and the inner boundary condition simply duplicates naturally. $R = 128a_1$ is chosen to be the outer radius. The radius of the second black hole is twice the size of the first one: $a_2 = 2a_1$. The radius of the first one is $a_1 = \sqrt{3}/2$ and the separation between both black holes is $b = \sqrt{12}$ in z-direction. The black holes have linear momenta. Black hole one gets $P_1 = (0, 0, -15)^T$ and black hole two gets $P_2 = (0, 0, 15)^T$. So, they model a head-on collision. These special parameters follow the work from [Muk96]. The calculation of the function $H(\mathbf{x})$ is not easy. With the results from Cook in [Coo91] an external calculation routine was written and implemented in `KASKADE`. Whether a little complicated, explicit formulas are posed in the appendix of Cook's paper. In principle, they rely on an infinite series which is aborted when the new term is small enough. In the special case considered in the present work, only about five terms of the infinite series were necessary to calculate. Nevertheless, the code slows down significantly.

Fig.(2.16) shows the domain which is now triangulated with `NETGEN` . We see two excised balls of different radii as well as some semi-transparent iso surfaces after one refinement step. This coarse grid solution is shown to demonstrate the triangulation of the grid generator. The domain is spherically cutted, so the outer boundary of Fig.(2.16) is only from artificial nature.

In Fig.(2.17) a plane cut of the symmetric solution is shown. Again, the conformal factor is encoded by color. Also we see that now, the adaptive refinement strategy is prior to the uniform one with respect to the energy error. The exact solution which appears in Eq.(2.17) is replaced with the finest numerical solution because no analytic solution exists anymore.

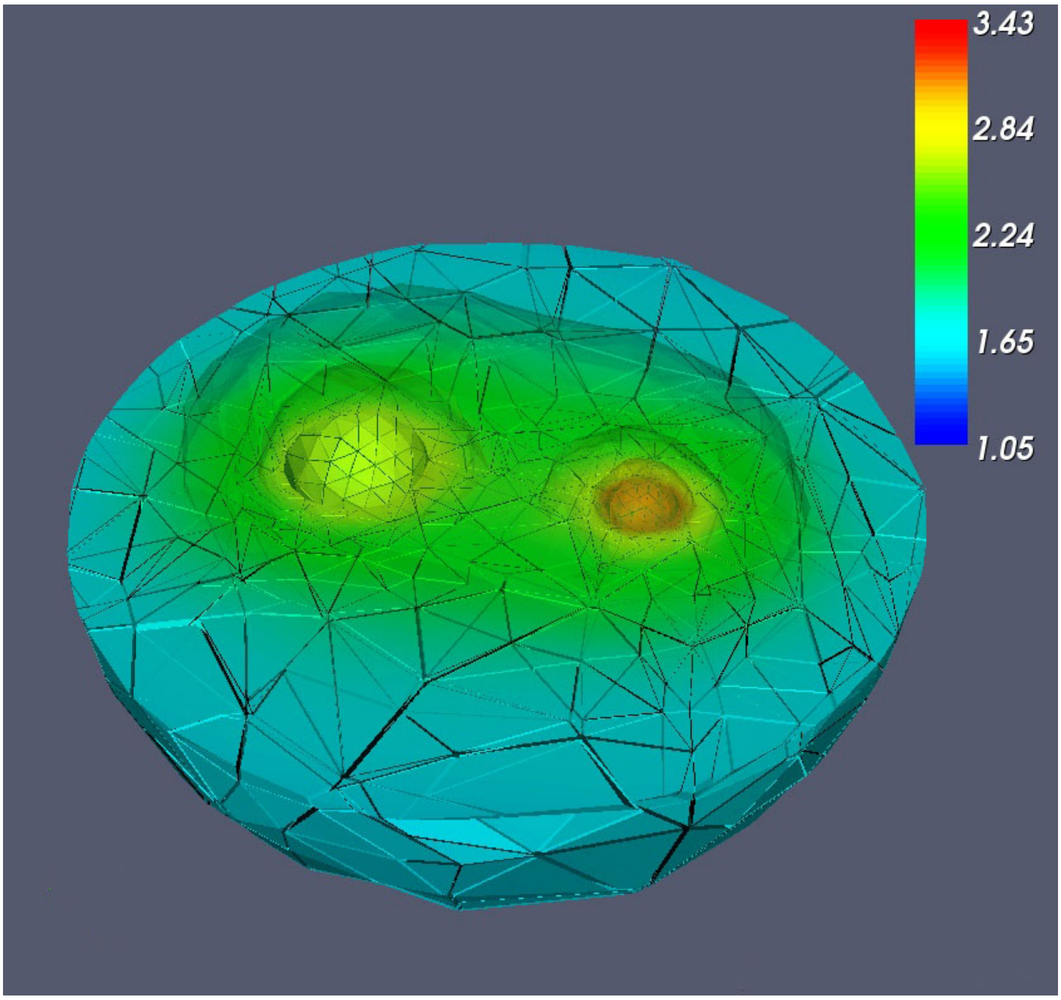


Fig. 2.16: Once refined triangulation of a NETGEN grid. Color encodes conformal factor. Three semi-transparent iso surfaces are shown as well.

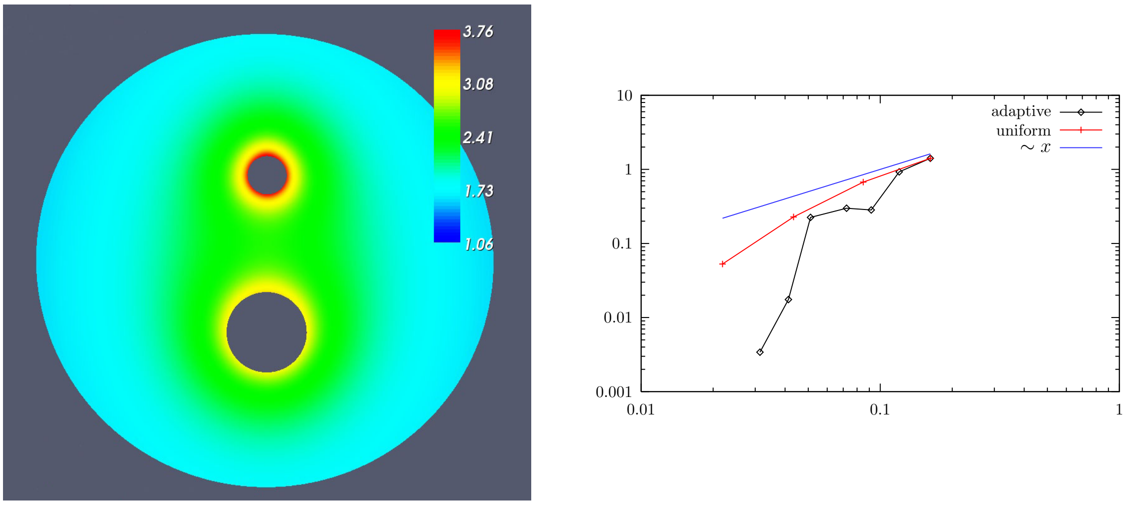


Fig. 2.17: Binary Black Hole Problem. *Left.* Solution with $\sim 10^6$ nodes and $\sim 4 \cdot 10^6$ tetrahedrons. Plane cut through axis-symmetric solution. Color encodes conformal factor. *Right.* Energy error of solution. x-axis: $h = 1/N_{\text{nodes}}$. y-axis: energy error.

CHAPTER 3

Time-Evolution of Einstein Equations with Regge Calculus

In 1961, Tulio REGGE published a paper [Reg61] where he introduced a highly interesting method to solve Einstein Equations, called *Regge Calculus*. It is a unique way of handling problems in General Relativity: a full four-dimensional calculus with an already built-in Finite-Element-like discretization scheme. From the Einstein-Hilbert action it directly yields an algebraic system of nonlinear equations. Furthermore it introduces primary variables which are completely free of coordinates. This independence of coordinates makes unique solutions possible.

The next section will give an introduction into Regge Calculus and the Time-Evolution scheme. Then, several simplicial decompositions for four dimensions are shown and employed numerical algorithms are presented. The chapter closes with results of my code developed from scratch which deal with problems of Numerical Relativity.

Recently, some works on Regge Calculus are able to give longterm results of the Time-Evolution for special problems. From importance is the successfully calculated Kasner spacetime solved in a parallel scheme without applying symmetries (see [GM97], [BGM$^+$97], [BG01]). This problem will be treated in this work as well, among many others. Before this, it was doubted that Regge Calculus could give any correct solution. Mainly, L. BREWIN discovered convergence rates of metrics according to Einsteins Equations which are equal to the convergence rate of an arbitrary metric (see [Bre95]). Concerning this, he posed modified equations which up to now were able to reproduce Schwarzschild spacetime in a $1+1$ setting (see [Bre98], [Bre97], [Bre01]). A successful approach to Schwarzschild spacetime without symmetries is still lacking. A very good review paper on Regge Calculus is [Gen04]. The early papers of R. SORKIN and M. GALASSI (see [Sor75], [Gal93]) are of paramount importance for this thesis and for Regge Calculus but lack results for longterm Time-Evolutions.

3.1 Theoretical Foundations of Regge Calculus

This section will show how the Regge Equations are derived. Then, these equations are specified to the Time-Evolution scheme introduced by R. SORKIN in [Sor75]. It was further developed by M. GALASSI in [Gal93] who introduces lapse and shift.

3.1.1 The Regge Action as an Approximation of The Einstein-Hilbert Action

Starting from Eq.(2.1), in Regge Calculus a simplicial decomposition of the domain is introduced first. Let Ω be the spacetime domain of interest, then Ω is decomposed in a set of N_s simplices s_i. This set of simplices represents the triangulation $\mathcal{T}$:

$$\Omega \approx \mathcal{T} = \bigcup_{i=1}^{N_s} s_i \,.$$

As a consequence, a spacetime domain with curved boundary can only triangulated approximately. In each simplex s_i the metric $g_{\mu\nu}$ is demanded to be flat. So, the global solution of the metric is *piecewise* flat in Regge Calculus. To summarize, the main approximation idea is to replace a curved spacetime with a piecewise flat spacetime. An easy illustration gives Fig.(3.1). Here, a 2-sphere (curved space) is replaced with an icosahedron (piecewise flat space). There are many ways to replace this 2-sphere with a piecewise flat space, therefore this step is ambiguous. The piecewise flat space expresses curvature with a new value which is paramount in Regge Calculus: The *deficit angle*. On the right hand side in Fig.(3.1) we can see that if the icosahedron is unrolled then one and the same edge will split up enclosing the deficit angle. Furthermore, one deficit angle corresponds to one point. In an arbitrary dimension d the geometrical object of the triangulation to which exactly one deficit angle corresponds is called *bone*. The bone is always of dimension $d-2$. In two dimension, the bones are points. In three dimensions the bones are edges and in four dimensions the bones are triangles. The picture of Fig.(3.1) holds true in an arbitrary dimension because for calculation of the deficit angle only the image space $\mathcal{M}$ is necessary. Here, the image space $\mathcal{M}$ is defined to be perpendicular to the bone. Since the bone is always $d-2$-dimensional, $\mathcal{M}$ is always two-dimensional. Considering General Relativity we have four dimensions.

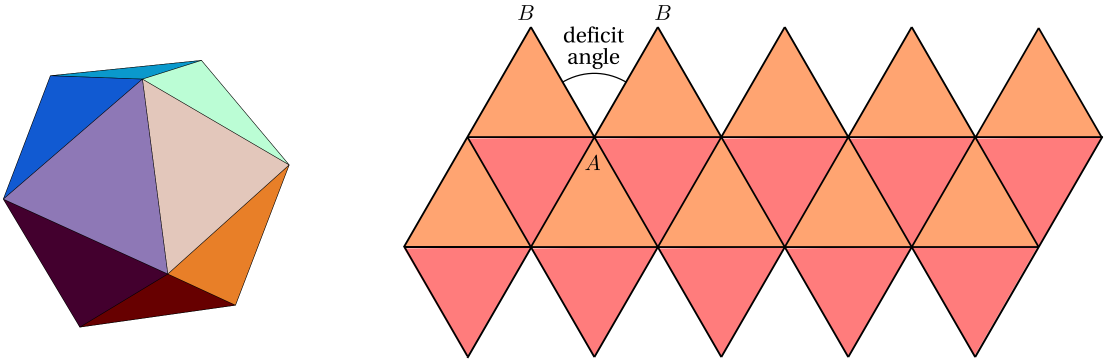


Fig. 3.1: *Left.* An icosahedron is a piecewise flat approximation of a 2-sphere. *Right.* An icosahedron unrolled in a flat reveals deficit angles which enclose an angle between edges which were one and the same before unrolling (e.g. edge $\overline{AB}$).

3.1.2 The Derivation of the Regge Action

In this chapter the Regge Action is derived in detail from the Einstein-Hilbert action. It follows the derivation from R. Sorkin in [Sor74]. After approximating the curved manifold with a piecewise flat manifold, the Einstein-Hilbert action in the form

$$S = -\frac{1}{2}\int_{\Omega} R\,\mathrm{d}V$$

can be evaluated. Since the manifold is flat everwhere, except at the bones, the contribution to curvature comes from the neighborhood of each bone. Furthermore, the result of a parallel transport of a vector around a closed loop is independent from parallel transports in the direction of the bone. Therefore the bone is homogeneous and the contribution of curvature is proportional to its area. Considering a three-dimensional triangulation for instance it becomes clear that a vector parallel transported not change if transported parallel to the edge. This is because the path on which the vector is parallel transported is part of one simplex assuming that this path is in the infinitesimal neighborhood of the edge.

For the exact derivation, only one bone is regarded. By means of the bone, the four-dimensional spacetime decomposes in the image space $\mathcal{M}$, the subspace orthogonal to the bone, and the bone space which is spanned by the bone. Regarding a timelike bone, the image space is euclidean and the bone space has Minkowski signature. Let us consider the standard basis for flat space: $\mathbf{x} = (t, x, y, z)^T$. Further, it is assumed, that the deficit angle vanishes in first place. Now, let us replace the cartesian coordinates of the image space with cylindrical coordinates: $\mathbf{x} = (t, x, r, \phi)^T$. Then the metric components are all zero except for the diagonal elements. They take the form $g_{tt} = -1$, $g_{xx} = 1$, $g_{rr} = 1$ and $g_{\phi\phi} = r^2$.

Next, the deficit angle ε is introduced. The origin of the coordinate system is the bone which is represented with a point in image space $\mathcal{M}$. Then, an auxiliary function $\exp(2\lambda(r))$ is introduced which smoothes out the defect in the way that the defect vanishes for small r but not for large r:

$$\begin{aligned}
&g_{rr} = 1 \qquad && g_{\phi\phi} = \exp(2\lambda(r)) = r^2 \qquad && \text{for small } r \\
&g_{rr} = 1 \qquad && g_{\phi\phi} = \exp(2\lambda(r)) = \left(1 - \frac{\varepsilon}{2\pi}\right)^2 r^2 \qquad && \text{for large } r
\end{aligned} \tag{3.1}$$

From the metric, the Christoffel symbols can be calculated, which do all vanish except:

$$\begin{aligned}
&\Gamma^{r}_{\phi\phi} = -\lambda' \exp(2\lambda) \qquad \text{and} \\
&\Gamma^{\phi}_{\phi r} = \lambda'
\end{aligned}$$

λ' denotes the partial differentiation of λ with respect to r. The only non-vanishing Riemann tensor component, the corresponding Ricci scalar and the value $\sqrt{-\det g}$ which is part of the curved infinitesimal volume element can be calculated to

$${R^{\phi}}_{r\phi r} = \lambda'' + (\lambda')^2 \qquad R = 2(\lambda'' + (\lambda')^2) \qquad \sqrt{-\det g} = \exp(\lambda)\,.$$

Applying the integration over image space $\mathcal{M}$ only yields

$$\begin{aligned}
-\frac{1}{2}\int_{\mathcal{M}} R\,\mathrm{d}V_{\mathcal{M}} &= -\frac{1}{2}\int_0^{\infty}\int_0^{2\pi} R\sqrt{-\det g}\mathrm{d}r\mathrm{d}\phi \\
&= -\frac{1}{2}\int_0^{\infty}\int_0^{2\pi} 2\exp(\lambda)''\mathrm{d}r\mathrm{d}\phi \\
&= -2\pi\int_0^{\infty}\exp(\lambda)''\mathrm{d}r \\
\text{with Eq.(3.1)} \quad &= -2\pi\exp(\lambda)'\big|_0^{\infty} \\
&= -2\pi\left(1-\frac{\varepsilon}{2\pi}\right) = \varepsilon\,.
\end{aligned}$$

$\mathrm{d}V_{\mathcal{M}}$ describes the curved volume element in image space. Via the integral, the auxiliary function $\lambda(r)$ vanished. To get the Einstein-Hilbert action of the piecewise flat spacetime we need only to integrate over the bone space. Since curvature vanishes aside from the bone, the integral over the bone space equals the area of the bone A_b. Therefore, the action becomes

$$S = -\frac{1}{2}\int_{\Omega} R\,\mathrm{d}V = \varepsilon A_b\,.$$

Allowing real and purely imaginary values, the derivation is analogous for a spacelike bone. In general, we have not only one but an arbitrary number of bones. Summarizing, the following equation holds:

$$S = -\frac{1}{2}\int_{\Omega} R\,\mathrm{d}V \overset{\text{piecewise flat}}{=} \sum_b \varepsilon_b A_b. \tag{3.2}$$

In the derivation, the Einstein-Hilbert action was modified with a factor of $-1/2$. It has no influence on the variational equations later, since an extremum is looked for. Eq.(3.2) is called the *Regge Action*. Each bone contributes to the action with an amount of $\varepsilon \cdot A$. Since the problem is posed in Minkowski geometry one has to distinguish between two cases:

(1) $\mathcal{M}$ involves a timelike direction, then: $\varepsilon \in i\mathbb{R}, A \in \mathbb{R}$,

(2) $\mathcal{M}$ involves *not* a timelike direction, then: $\varepsilon \in \mathbb{R}, A \in i\mathbb{R}$.

If $\mathcal{M}$ inhabits no timelike direction, then the deficit angle is real valued and is calculated in Euclidean geometry. If the space spanned by the bone has no timelike directions, then the area of the bone is real valued. Otherwise one has to deal with pure imaginary values of the deficit angle and the area of the bone respectively. The algorithm for calculation will be discussed in detail later in section 3.3.

Brief Formulas for the Deficit Angle and the Area of the Bone

The deficit angle ε_b is calculated by summing up the *circulation angles* α_i of each simplex, hinging on the bone b. That means each simplex s with $b \subset s$ is needed for the calculation. This is illustrated in Fig.(3.2). Then the deficit angle is

$$\varepsilon_b = \begin{cases} 2\pi - \sum_{i|b\subset S}\alpha_i & \mathcal{M}\text{ is euclidean} \\ -\sum_{i|b\subset S}\alpha_i & \text{else}\,. \end{cases}$$

If we assume a two-dimensional test problem, the angles α_i can be calculated directly. For higher dimensions, the angles α_i are not directly accessible but must be calculated with an appropriate method. In this thesis the method from section 3.3.2 is applied regarding the four dimensions of General Relativity.

The area A of the bone b is calculated with a formula for Euclidean geometry. Let x^2, y^2 and z^2 be the squared edge length of each edge of the bone, then,

$$A_b = \frac{1}{4}\sqrt{2(x^2y^2 + x^2z^2 + y^2z^2) - x^4 - y^4 - z^4}\,, \tag{3.3}$$

is the area of the bone.

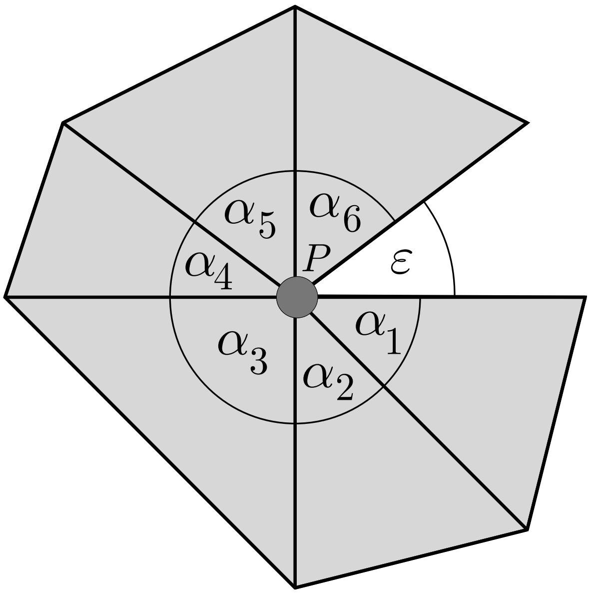


Fig. 3.2: The deficit angle is calculated by adding $-\alpha_i$ subsequently. If image space $\mathcal{M}$ is a spacelike bone, then 2π is added additionally. P is the bone and is represented with a point in the image space. For an arbitrary dimension, curvature always resides in the bone and the bone is always represented with a point in the image space.

3.1.3 The Regge Equations

From Eq.(3.2) the Regge Equations can be derived by varying this formula with respect to a set of variables. This is analogue to the derivation of Einstein equations or the ADM equations as shown in chapter two. Here, we can choose already a finite set of variables instead of choosing a set of functions like the metric. This yields directly a system of equations. This is analogous to the FEM with an already finite function space. This is because Regge Calculus was already discretized in the first step with introducing a triangulation. Nevertheless, care must be taken in choosing this set of variables. The triangulation must have no freedom after variation. A natural choice are the edge lengths. By fixing all edge lengths the triangulation is fixed. So, the set of variables is complete. But unfortunately the edge length can become purely imaginary when the associated scalar product is negative in case of the edge being timelike. So instead, we choose the *squared edge lengths* as the set of variables. The variation gives

$$G_i = \sum_b \varepsilon_b \cdot \frac{\partial A_b}{\partial l_i^2} + \sum_b \frac{\partial \varepsilon_b}{\partial l_i^2} \cdot A_b \,. \tag{3.4}$$

Here, G_i denote the N_e Regge Equations, where N_e is the total number of the edges. l_i^2 is the squared edge length. It is worth to note that due to this variation there is an one-to-one correspondence between a specific edge i and its equation G_i. Each edge gives rise to exactly one equation.

T.Regge has shown in [Reg61] that the latter term of Eq.(3.4) vanishes. So,

$$G_i = \sum_b \varepsilon_b \cdot \frac{\partial A_b}{\partial l_i^2} \,.$$

The derivative of A_b with respect to l_i^2 will only be non-vanishing if the edge i is part of the bone. So the sum can be restricted to bones which have edge i in common. Now, let l_i^2, l_j^2 and l_k^2 be the squared edge lengths of the edges of such a bone, then:

$$\frac{\partial A_b}{\partial l_i^2} = -\frac{1}{16 A_b}(l_i^2 - l_j^2 - l_k^2) \,.$$

This follows directly from Eq.(3.3). Let e_i be the edge with index i and $b(ijk)$ denote the bone with edge indices i,j and k, then the Regge Equations are finally:

$$G_i = -\frac{1}{16} \sum_{b(ijk) \supset e_i} \varepsilon_b \cdot \frac{1}{A_b} \cdot (l_i^2 - l_j^2 - l_k^2) \,, \quad i = 1, 2, \ldots, N_e \,.$$

At this place, source terms T_i representing the energy momentum tensor enter. The equations would read $G_i = T_i$ but in this work only the vacuum case is investigated: $G_i = 0$. Again, i runs over all edges in the triangulation.

Then, if N_e is the number of edges with unknown length, the number of unknowns and the number of equations are equal at this point. Now, we can try to solve this non-linear problem by means of a Newton-Raphson method for instance. The determination of the deficit angle was not discussed till now but an algorithm for calculation is stated in section 3.3.2. Unfortunately, the dimension of this system is very high since there are N_e equations and N_e unknown squared edge lengths for which to solve. Rudimentary tests showed convergence for low N_e but at around $N_e \approx 50$ the applied Newton scheme lacks convergence. Another issue is that the amount of time needed to solve the problem increases fast because the problem is four-dimensional. Increasing spatial and time resolution let the performance drop. On the other hand, the local structure of Regge Equations yield a global sparse matrix. Therefore efficient methods exist to solve such a system. But then, the solution at the first *and* the last hypersurface is necessary, as well as the solution on the spatial boundary. This may not be of interest in physical problems.

Apart from that, it is more reasonable to introduce a $3+1$ scheme analogous to the ADM formalism. Like argued in chapter two, this idea splits up the problem in first, calculating initial data and second, employing a Time-Evolution scheme. R. Sorkin developed a Time-Evolution scheme in [Sor75]. There, the global system with a $N_e \times N_e$ matrix decomposes in a lot of local 15×15-systems. This scheme will be developed in the next part.

3.1.4 The Time-Evolution Scheme

In this section the Time-Evolution Scheme first posed by R.Sorkin in [Sor75] is derived. In contrast to triangulate a spacetime first and then solve the resulting Regge Equations globally, R.Sorkin starts from initial values between two consecutive hypersurfaces and then enhances the simplex structure in timelike direction. This gives us many local and independent quadratic systems of a small number of equations (15 or 7 in the work from Regge).

The Time-Evolution Idea of R.Sorkin

The Time-Evolution Idea is visualized in a two-dimensional sketch in Fig.(3.3) and Fig.(3.4). First, initial values between two consecutive hypersurfaces are given: $\mathcal{T} = \mathcal{T}_0$. This is analogous to a thin-sandwich formulation. All Regge Equations corresponding to the initial triangulation are assumed to be fulfilled. Second, a new point A^+ is added to $\mathcal{T}$. Also edges and simplices connecting this new point A^+ with $\mathcal{T}_0$ are added to $\mathcal{T}$. With this enhancement of the triangulation new equations arise, corresponding to edges with $e_i \in \partial\mathcal{T}_0$ but $e_i \notin \partial\mathcal{T}$. The set of these edges is denoted with E. There is also the set of new edges U, (edges connecting A^+ with $\mathcal{T}_0$) which are the unknowns corresponding to point A. In the triangulation based on $\mathbb{R}^3$ constructed by R.Sorkin the number of new edges and new equations are equal: $|U| = |E| = 15$. But also for all triangulations investigated this number is around 15 and the equality $|U| = |E|$ holds. Especially, this equality is guaranteed with the prism method which is discussed later in section 3.2.2.

Unfortunately, as illustrated in Figure Fig.(3.4), it is obvious that one has to fix three of the edges in U so that it is geometrically clear where the point A^+ is located. In a two-dimensional example two edges have to be fixed and in four dimensions four edges have to be fixed. These fixed edges correspond to lapse and shift in ADM formalism. The edges which fix point A^+ in the geometry are called *lapse edges* if the edge is timelike and *shift edges* if the edge is spacelike. Normally, the edge $\overline{AA^+}$ is the only timelike edge and is used to be the one *lapse edge*. Via these quantities we can control the evolution of point A like lapse and shift can do in ADM formalism. As a drawback the formerly quadratic system ($|U| = |E|$) has become rectangular ($|U| - 4 \neq |E|$).

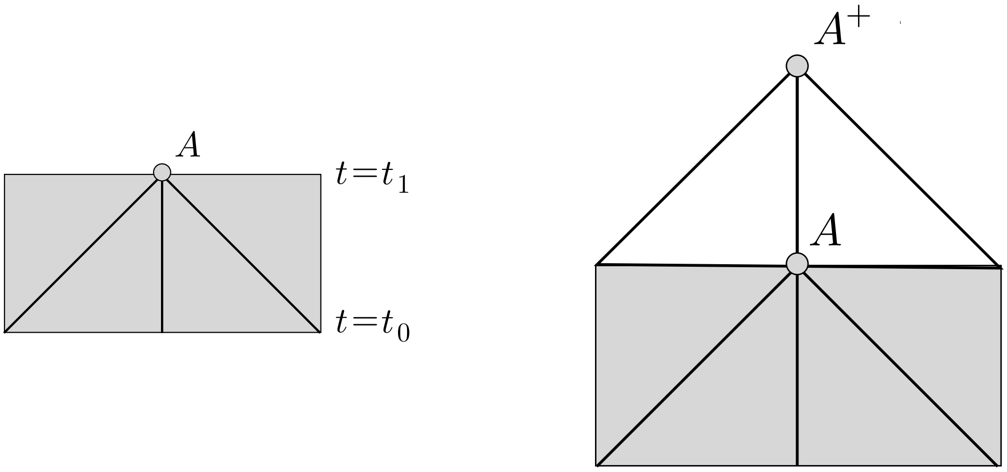


Fig. 3.3: *Left.* The initial values between two hypersurfaces at time $t = t_0$ and $t = t_1$ are given. Point A is going to be evolved. *Right.* The triangulation is enhanced by the point A^+ on the new hypersurface, as well as new simplices and new edges having A^+ in common.

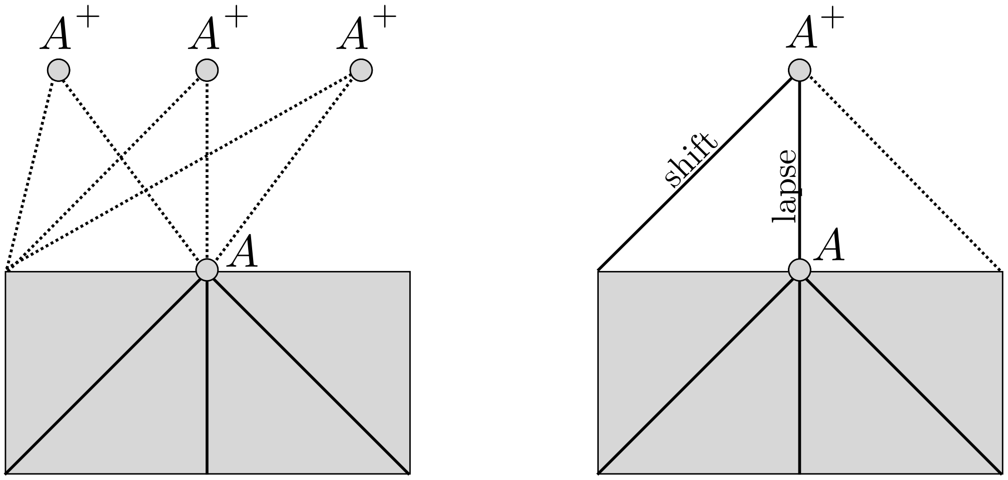


Fig. 3.4: *Left.* If no new added edges are fixed, the problem stays ambiguous. Point A^+ is not located. *Right.* By fixing some of the new added edges (2d: 2, 4d: 4), point A^+ is fixed and the problem is well posed.

Parallelization – Graph Coloring

The described procedure of Time-Evolution for one single point has to be applied for each point of the spacetime at $t = t_1$. In general the scheme is parallelizeable. By applying *graph coloring* methods to the hypersurface at $t = t_1$, points which can be evolved independent from each other can be determined. Given the set $\mathcal{P}$ which contains all points at $t = t_1$, a decomposition into distinct sets Q_i is looked for. Each set Q_i should only contain points which are sufficient far away from each other. In Regge Calculus that means, each two points of any set Q_i are not connected with an edge:

$$\begin{aligned} \mathcal{P} &= \bigcup_i Q_i\,, && \text{Points in } Q_i \text{ are not connected with an edge} \\ Q_i \cap Q_j &= \{\,\} && \text{for } i \neq j \end{aligned}$$

This approach is sketched in Fig.(3.5). In this two-dimensional example where space has only one dimension each second point can be evolved in parallel. Here, the points labeled with A would give the point set Q_1 and all points labeled with B would give the point set Q_2. Therefore, $P = Q_1 \cup Q_2$ holds. In general each point of the set Q_i can be evolved in parallel. The sets Q_i can be ordered in arbitrary way. Applied to the example it would mean that it does not matter if we evolve the A-points first and than the B-points or vice versa.

Problems with Two-Dimensional Illustrations

Perceiving two-dimensional sketches gives a picture of Regge Calculus and its ideas. It is full satisfactory for many problems, for instance the problem of calculating the deficit angle reduces in any dimension to a two-dimesional problem which can be fully illustrated with a two-dimensional sketch.

Contrary, when sketching triangulations to illustrate the Time-Evolution idea for example, it has to be kept in mind that the complexity of three- or four-dimensionsal problems is

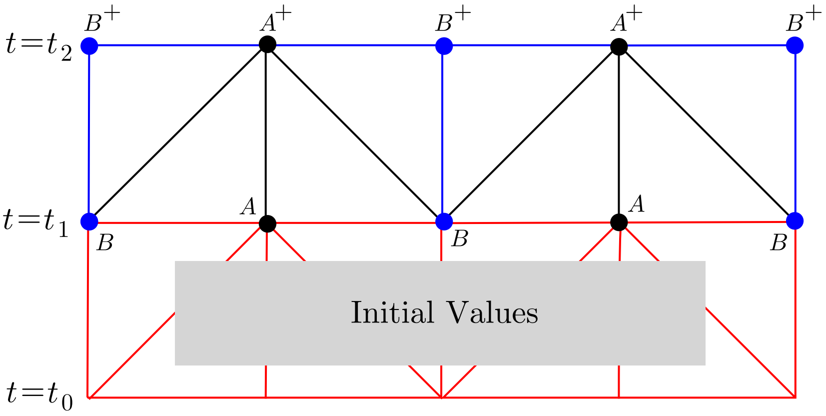


Fig. 3.5: Points labeled with A can be evolved in parallel (black lines). Afterwards, points labeled with B can all be evolved in parallel (blue lines). Here, boundary issues are not discussed.

higher, qualitatively and quantitatively. The following example illustrates this. In Fig.(3.5) we see that the two-dimensional triangulation is based on squares. The triangulation of a two-dimensional square is trivial. On the other hand a triangulation of a cube in three or four dimensions is non-trivial. New qualitative aspects occur. In two dimensions it is obvious that the spacetime between $t = t_1$ and $t = t_2$ is filled correctly with triangles. But in three and four dimensions there are triangulations which are incompatible with the Time-Evolution idea as shown later in section 3.2.1. This fact can not be perceived from Fig.(3.5). To conclude, it is worth doubting that obvious relationships in two-dimensional sketches hold also in three and four dimensions.

Summarizing the Time-Evolution Scheme

In four dimensions we have the following procedure: For each point p at $t = t_1$ the simplex structure is enhanced. Due to this procedure N_{eq} new equations occur which became inner edges due to the enhancement of the triangulation. There are also N_{edges} new edges in the enhanced triangulation which edge lengths are unknown.

The new point A^+ is not fixed in the geometry because all edge lengths emanating from it are unknown. Therefore, four conditions must be be stated which fix the point in the geometry. That is in general one lapse condition and three shift conditions. These conditions are in general chosen specific to the problem. Now, there are N_{eq} equations and $N_{\text{var}} = N_{\text{edges}} - 4$ unknowns left. For all tested triangulations it is $N_{\text{eq}} = N_{\text{edges}} = N_{\text{var}} + 4$. So, in general an overdetermined system of non-linear equations must be solved.

3.1.5 Numerical Tools for Solving Regge Equations

Solving a non-linear system of equations is non-trivial. For one equation and one unknown there exist many approaches. For more than one dimension the Newton-Raphson method is often the best choice. It has the best convergence when the start value is near to the root but lacks reliability if the start value is far away.

In the Time-Evolution scheme there is already a restriction to the start values and all iterated values later: The triangle inequality. In four dimensions there are ten edges per simplex. So, for every three edges of the ten, the triangle inequality must hold. The equality must be modified according to the signature of the plane containing the triangle. This is described with the timelike direction. If the plane contains a timelike direction, then the signature is $-+$, else the signature is $++$. In the used code a failure of the triangle inequality in Minkowski space will be automatically detected by the used algorithm. Furthermore, the algorithm will provide us with the signature of the metric with any given ten squared edge lengths. This will be discussed later in section 3.3.

In the next section the Newton-Raphson method will be introduced which transfers a non-linear to a linear system of equations. The next two sections will shortly deal with the problem of how to solve the linear system of equations and make a distinction between quadratic and rectangular systems.

The Newton-Raphson Method

Nowadays, this method is standard. For instance, [DH03] gives a good introduction and [Deu06] covers a lot of cases in complete detail. Therefore a short introduction should be sufficient. In one dimension the problem reads,

$$f(x) = 0\,. \tag{3.5}$$

Let x^0 denote the start value, $\tilde{x}$ denote the unknown solution and $\Delta x = \tilde{x} - x^0$. Then the Taylor expansion gives,

$$\begin{aligned} 0 = f(x^0 + \Delta x) &= f(x^0) + f'(x^0)\Delta x + \mathcal{O}(|\Delta x|^2)\,, && (3.6)\\ \Rightarrow \quad f'(x^0)\Delta x &\approx -f(x^0)\,. && (3.7)\end{aligned}$$

Therefore if $f'(x^0) \neq 0$ the approximated solution is

$$\tilde{x} \approx x^1 = x^0 - \frac{f(x^0)}{f'(x^0)}\,. \tag{3.8}$$

Replacing $1 \to k+1$ and $0 \to k$ in Eq.(3.8) the Newton iteration equation is complete. It can be shown that near to the exact solution the scheme has quadratic convergence, that means

$$|\tilde{x} - x^{k+1}| \leq C \cdot |\tilde{x} - x^k|^2\,.$$

In higher dimensions the scheme does not change much. It is in general applicable to the case of m equations and n unknowns. In Eq.(3.7) the derivative f' must be replaced with a $m \times n$-dimensional Jacobi matrix J. Δx becomes an n-dimensional vector and the right hand side $-f(x^0)$ transfers to an m-dimensional vector. Then the resulting system of equations

$$J\,\Delta\mathbf{x} \,=\, \mathbf{rhs} \tag{3.9}$$

is linear. Eq.(3.9) is solvable with the LU method or QR method respectively. These methods are discussed in section 3.4.3. Often, the matrix J is not applicable analytically or it takes too much effort to calculate it. To circumvent this problem, the matrix components can be approximately calculated with finite Differences. Depending on the needed accuracy we can take one of both differences:

$$\frac{\partial f_i}{\partial x_j} = \frac{f_i(x_j + h) - f(x_j)}{h} + \mathcal{O}(h) \,, \tag{3.10}$$

$$= \frac{f_i(x_j + h) - f(x_j - h)}{2h} + \mathcal{O}(h^2) \,, \tag{3.11}$$

Taking less accurate Eq.(3.10) gives a large speedup, since $f(x)$ was already calculated for the right hand side. Contrary, the equation Eq.(3.11) provides us with a better approximation but takes twice the time for calculation. Differenet methods are applicable at this point if there is need for it. First, we can calculate J only once in the first iteration step and then keep it constant. Second, J is calculated once but updated with an effort-less method like suggested by C.G. BROYDEN in [Bro65]. Third, to cancel the free parameter h, automatic differentiation can be applied. In this work it is sufficient to fix parameter h by testing.

Paramount to the Newton scheme is the *termination criterion.* When should the iteration stop? P.DEUFLHARD described in [Deu06] Newton methods in detail and derived termination criterions for each specific case. They depend on a lot of factors and must be specified in each special case. Of course, another question is if the Newton method does converge at all. Practically, test runs can be undertaken with known solutions. For more convenience, convergence monitors can be implemented, as suggested in [Deu06] . If they fail there is also plenty of theoretical framework to investigate the reason of failure and methods to improve the ordinary scheme. For instance, a simple damping scheme can be employed.

A Simple Damping Scheme for the Newton Method

From a one-dimensional example it is obvious that the Newton method can fail in a lot of cases. If the start value is too far away from the root of interest it will give another root in the end or the scheme does not converge at all. In practice, a bad iteration step can be detected when $|f(x_{k+1})| > |f(x_k)|$. Here, f is the function in Eq.(3.5) and x_k are the iteration values. In this case there are many possibilities to replace the bad x_{k+1} with a better one, $\tilde{x}_{k+1} \in [x_k, x_{k+1}]$, so that then $|f(\tilde{x}_{k+1})| < |f(x_k)|$. For instance, the *line search method* makes use of polynomial interpolation to calculate a better $\tilde{x}_{k+1}$. In this work the line search method was as reliable as a much simpler scheme. Here, it is sufficient to half the step size as long as necessary. This is equivalent to find the lowest positive integer $q \geq 0$ for which,

$$\tilde{x}_{k+1} = x_k + 2^{-q} \cdot (x_{k+1} - x_k) \ , \qquad \text{with } |f(\tilde{x}_{k+1})| < |f(x_k)| \,. \tag{3.12}$$

Fig.(3.6) illustrates three cases where the ordinary Newton method fails. Let $x(A)$ denote the x-value of point A, then after one Newton step $x(A) \rightarrow x(A+)$ the value is in the region around another root and therefore the Newton scheme fails. The point B gives a steeper tangent but the Newton scheme $x(B) \rightarrow x(B+)$ also puts the value in the region around another root. But here this failure can be detected because $|f(x(B+))| > |f(x(B))|$ and the damping scheme from Eq.(3.12) can be applied. After two steps $x(B+) \rightarrow x(B_1) \rightarrow x(B_2)$ this satisfies $|f(x(B_2))| < |f(x(B))|$ *and* point B_2 is again in the region around the right root.

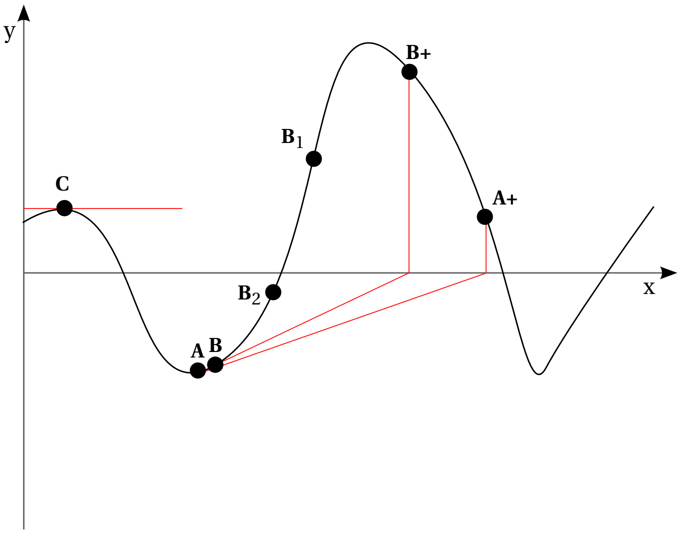


Fig. 3.6: Three cases where ordinary Newton method fails. (A) The tangent is too flat and pushes the approximate solution to another root. (B) The tangent is too flat but since $|f(B+)| > |f(B)|$ a damping scheme can be activated yielding a good result after two damping steps B_1 and B_2. (C) Since $f'(C) = 0$ the Newton scheme fails immediately.

3.2 Simplicial Decompositions of the Four-Dimensional Domain

As shown in section 3.1.1, the first step to derive the Regge Action Eq.(3.2) from the Einstein-Hilbert action is to triangulate the given domain. This section will deal with the construction of a triangulation. There is not one unique way but many different. Although this section gives no complete introduction into the broad field of triangulation it presents the special triangulations used in the code.

Triangulations can directly be percieved in two dimensions. In three dimensions only a restricted visual understanding is possible because one has to deal mostly with two-dimensional projections. This problem increases when considering four-dimensional triangulations. Therefore a first natural step towards triangulations in four dimensions is to simplify the problem. In this work, only simplicial triangulations are considered.

A first approach, discussed in the next section, is to consider a decomposition of the domain into cubes. Then, each cube is triangulated into simplices with a given scheme. So, we have to deal with cube pavements only, which is easier than to work with single simplices. Also, for the triangulation of a cube into simplices no additional nodes are necessary. The only additional effort comes from the faces of the cubes. All faces of touching cubes must be triangulated identically to give a valid triangulation. The term *valid triangulation* was defined in Eq.(2.15).

A second approach is to make use of the $3+1$ decomposition of spacetime. Given a triangulation on two consecutive hypersurface which are both three-dimensional and identical, the so called *prism method* is able to fill the four-dimensional region of spacetime between these two hypersurfaces with 4-simplices. This method will be discussed in the second part of this section.

The last part will present a triangulation of a closed universe which turns out to work very well in practice. A closed universe is a domain of $T^3 \times \mathbb{R}$ where T^3 is a 3-torus. The triangulation was first presented by GENTLE and MILLER in [GM97].

3.2.1 Triangulations of the 4-Cube

In section 2.3.1 methods to triangulate the 3-cube where presented. Here, the 4-dimensional case is discussed. Results from the 3-dimensional case can often be applied here too. First, the triangulation of P.S. MARA is presented. It needs the lowest number of simplices to construct a 4-cube. Second, Kuhn's method, introduced for three dimensions in section 2.3.1, is applied to four dimensions.

The Triangulation of P.S. Mara

Patrick Scott MARA published in [Mar76] an important triangulation of a 4-cube. It consists of 16 simplices only. He proved, that this is the lowest possible number of simplices to construct a 4-cube. Such triangulations are called *minimal triangulations.* There is active research on d-dimensional minimal triangulations as [BS05] show. The idea to construct MARA's triangulation starts with the same method of triangulating the 3-cube as shown in section 2.3.1 and as sketched in Fig.(2.9). Taking the 4-cube we chip off vertices of a simplex. As an effect, these vertices only appear in one simplex of the triangulation.

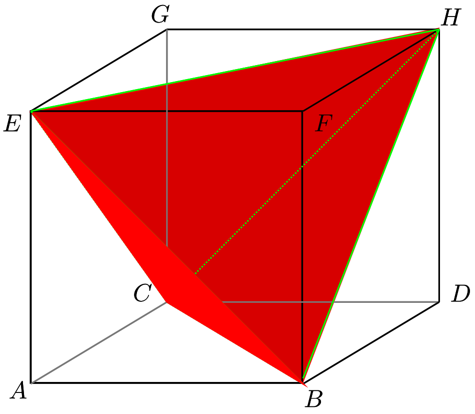


Fig. 3.7: Triangulating a 3-cube by chipping off the cube vertices A,D,G and F. Then the remaining domain (red) is non-empty and constitutes the simplex $ECBH$. The same idea is applied to get MARA's triangulation of the 4-cube.

From Fig.(3.7) we see naturally which vertices can be chipped off. Let us consider the unit cube and let us give coordinates to each vertex according to Tab.(3.1).

	$(x_0\ x_1\ x_2\ x_3)$		$(x_0\ x_1\ x_2\ x_3)$
A	0 0 0 0	A^+	1 0 0 0
B	0 0 0 1	B^+	1 0 0 1
C	0 0 1 0	C^+	1 0 1 0
D	0 0 1 1	D^+	1 0 1 1
E	0 1 0 0	E^+	1 1 0 0
F	0 1 0 1	F^+	1 1 0 1
G	0 1 1 0	G^+	1 1 1 0
H	0 1 1 1	H^+	1 1 1 1

Tab. 3.1: Point labeling for the 4-cube.

Then, vertices whose coordinate sum $\sum x_i$ is even can be chipped off. This is vertex A, D, F, G, B^+, C^+, E^+ and H^+. Then, the remaining domain, consisting only of vertices with odd coordinate sum, can be filled with eight simplices. One feature of the simplices which where constructed by chipping off vertices is, that they completely shape the boundary. So, it is natural to call them *outer simplices*. All other simplices do not have any face on the boundary of the cube. So, they are called *inner simplices*. With this terminology the 4-cube is decomposed in 16 simplices as shown in Tab.(3.2).

With this minimal triangulation, the number of edges is also the smallest possible. This is an advantage since a smaller computational effort per cube can be expected. Unfortunately, the idea of Time-Evolution (that means introducing a point, connecting it with an edge, looking for simplices hinging at the edge, ...) is not applicable to this triangulation. This is because of the inner simplices in the triangulation which consist not of any cube edge but only of diagonals (2-diagonals, 3-diagonals, 4-diagonals). The issue can be understood by means of Fig.(3.7). If we apply the idea of Time-Evolution to triangulations with inner simplices, then the region between two hypersurfaces are not paved completely with simplices. This situation can be seen in the figure where the red simplex only consists of diagonals. So, by only looking for simplices which hinge on a cube edge, which is the case in the Time-Evolution scheme, this kind of simplex never will be added to the triangulation. Rather, after a timestep, the

Outer Simplices		Inner Simplices	
1	$ABCEA^+$	1	BG^+CEA^+
2	$DBCHD^+$	2	$BG^+EA^+F^+$
3	$FBEHF^+$	3	$BG^+CA^+D^+$
4	$GCEHG^+$	4	BG^+CEH
5	$B^+A^+D^+F^+B$	5	$BG^+A^+D^+F^+$
6	$C^+A^+D^+G^+C$	6	BG^+EHF^+
7	$E^+A^+F^+G^+E$	7	BG^+CHD^+
8	$H^+D^+F^+G^+H$	8	$BG^+HD^+F^+$

Tab. 3.2: MARA's triangulation of the 4-cube with 16 simplices. 8 simplices have faces on the boundary of the cube (*Outer Simplices*), 8 simplices have not (*Inner Simplices*).

triangulation between two hypersurfaces look like a swiss cheese. The holes represent the inner simplices which were not added.

To summarize, the presented Time-Evolution scheme can be applied only to triangulations of a cube without inner simplices. Of course it is possible that a modified version can be applied. There is no reason why to exclude any triangulation in principle. In this work no further investigation on this subject was made.

Kuhn's Method in Four Dimensions

KUHN's method is easy to apply in any dimension. It was introduced in section 2.3.1. Also, we have no problem with stitching together the triangulated cubes because the touching faces are always triangulated identically. In the following graph we go each possible way along edges from point A to H^+. The point labeling of the latter section from Fig.(3.1) is used. The one restriction for the way from A to H^+ is that the next point coordinate along the way has to increase. That means for instance, that the path $F \rightarrow E$ is forbidden but $E \rightarrow F$ is allowed. Fig.(3.8) shows the building scheme for the 24 simplices which triangulate the 4-cube. It is known that triangulations by KUHN's method need the highest number of simplices. Such triangulations are called *maximal triangulations*.

3.2.2 The Prism Method in Four Dimensions Using Arbitrary 3D Grids

The prism method is powerful since it triangulates the domain $\Omega = S \times [0,1]$, where S is an arbitrary $n-1$-dimensional triangulation. The prism method can also generate triangulations of a cube which by the way may coincide with other approaches. This depends on S and the way the points on S are ordered in the upcoming scheme. Let t denote the new coordinate. One feature of the prism method is that on $t = 0$ and $t = 1$ the domain Ω has the identical triangulation S. For the sake of simplicity let us restrict the problem to four dimensions. So, each point $p = (0, p_0, p_1, p_2)$ on a hypersurface has its pendant described by $p^+ = (1, p_0, p_1, p_2)$ on the consecutive hypersurface. Let $\mathbb{T}$ denote the triangulation of the four dimensional domain. Then the algorithm 3 represents the prism method in general.

When generating the triangulation, the set $\mathbb{M}$ gives information whether the incomplete triangulation has already points of the same spatial coordinates at $t = 1$ (point marked with symbol $+$). If yes, then this $+$-point is used to generate the simplex, the corresponding point at $t = 0$ is skipped. The algorithm shows that each prism is triangulated by its own almost independently. But we have to take care that matching faces are triangulated in the same

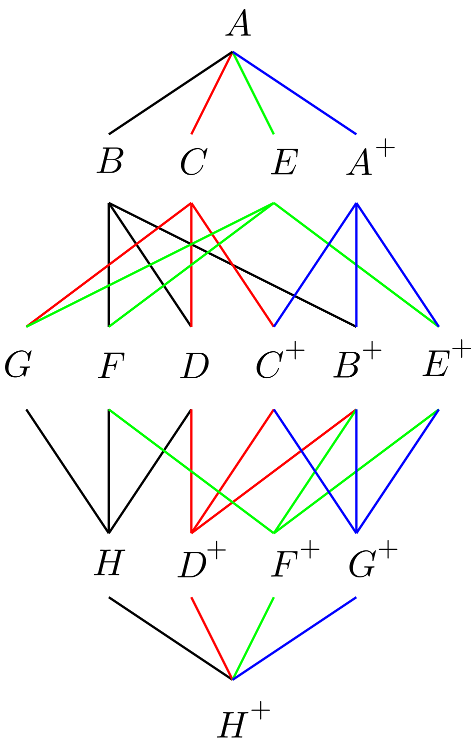


Fig. 3.8: Scheme for constructing the 24 Kuhn simplices. The colored lines have no separate meaning but help to follow the path. The first simplex would be for example $ABGHH^+$ and the last $AA^+E^+G^+H^+$. All simplices have the four-diagonal AH^+ in common.

Algorithm 3 Prism method in four dimensions.

```
$\mathbb{T} = \{\}$
$\mathbb{M} = \{\}$
for all tetrahedrons $s_i \in S$ do
    for all points $p_j \in s_i$ with respect to point order do
        Let $p_{j0}$, $p_{j1}$ and $p_{j2}$ denote the remaining points of $s_i$.
        if $p_{j0} \in \mathbb{M}$ then
            $p_{j0} := p_{j0}^+$
        end if
        if $p_{j1} \in \mathbb{M}$ then
            $p_{j1} := p_{j1}^+$
        end if
        if $p_{j2} \in \mathbb{M}$ then
            $p_{j2} := p_{j2}^+$
        end if
        $\mathbb{T} = \mathbb{T} + \text{simplex } \left\{p_j, p_j^+, p_{j0}, p_{j1}, p_{j2}\right\}$
        $\mathbb{M} = \mathbb{M} + p_j$
    end for
    $\mathbb{M} = \{\}$
end for
```

way (valid triangulation). This is guaranteed with a fixed order of the points p. For each tetrahedron this point order must be respected. This method is illustrated with a simple example. Let $n = 3$ and S the triangulation of a square as shown in Fig.(3.9).

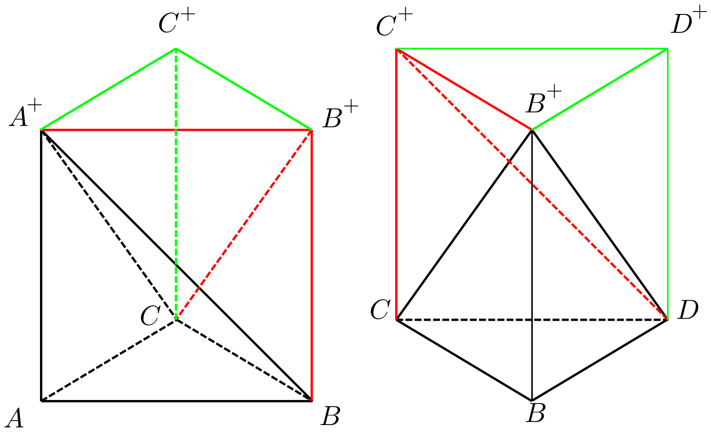


Fig. 3.9: Scheme for constructing a 3-cube from a 2-cube (square). The result after one step is shown for each prism over triangle ABC and BCD. Both prisms can be constructed in parallel.

S exists of two triangles. First, $\triangle ABC$ and second, $\triangle BCD$. The integer coordinates are: $A = (0,0), B = (0,1), C = (1,0)$ and $D = (1,1)$. We take the lexicographic point order. Now, both prisms can be constructed in parallel. Fig.(3.9) shows the edges which are added. The first step is visualized with black color, the second with red and the last with green color. We can see that in each step the active point is connected by a perpendicular edge with its $+$-pendant. The active point and its $+$-pendant are part of the 3-simplex. The other two nodes come from the triangle. If these two nodes have $+$-pendants already, then they are used. Otherwise the points on the basis are used. Following Fig.(3.9) a detailed explanation of how the prism over triangle ABC is built is given here:

(0) In the beginning only points A,B,C exists. $\mathbb{M} = \{\}$.

(1) Point order says point A is the first one. A^+ is added. Points B and C are considered. Because $B \notin \mathbb{M}$ and $C \notin \mathbb{M}$ the simplex $s_1 = \{A, B, C, A^+\}$ is added. Now: $\mathbb{T} = \{s_1\}$, $\mathbb{M} = \{A\}$.

(2) Point order says point B is next. B^+ is added. Points A and C are considered. $A \in \mathbb{M}$ and $C \notin \mathbb{M}$, therefore $s_2 = \{B, A^+, C, B^+\}$ is added. Now: $\mathbb{T} = \{s_1, s_2\}$, $\mathbb{M} = \{A, B\}$.

(3) Point order says point C is last. C^+ is added. Points A and B are considered. $A \in \mathbb{M}$ and $B \in \mathbb{M}$, therefore $s_3 = \{C, A^+, B^+, C^+\}$ is added. Now: $\mathbb{T} = \{s_1, s_2, s_3\}$, $\mathbb{M} = \{A, B, C\}$.

(4) Triangulation of prism over triangle ABC complete.

According to this scheme six tetrahedrons shown in Tab.(3.3) are constructed.
The point order $A \rightarrow B \rightarrow C \rightarrow D$ was important to construct a valid triangulation. Let us have a look at the only face where both prisms touch each other: The quadrangle BCB^+C^+. One necessary and sufficient condition on the triangulation is, that there exists only one of both diagonals BC^+ or B^+C respectively as an edge in the triangulation. In the example only B^+C exists. So the triangulation is valid. If both diagonals would be part of the triangulation then there would exist two triangles whose intersection is again a triangle. This would result in an invalid triangulation. The result of the prism method is shown in Fig.(3.10).

Alternatively, a modified prism method in four dimensions can be applied. It can be constructed analogous to section 2.3.1 where the approach is discussed in three dimensions.

simplex	tetrahedrons			
ABC	A	B	C	A^+
	B	C	A^+	B^+
	C	A^+	B^+	C^+
BCD	B	C	D	B^+
	C	D	B^+	C^+
	D	B^+	C^+	D^+

Tab. 3.3: The prism method used on two triangles to construct a triangulation of the 3-cube. The point order is $A \rightarrow B \rightarrow C \rightarrow D$

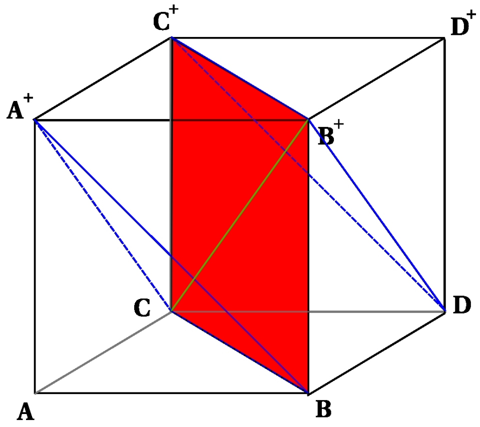


Fig. 3.10: Finished prism method. Color indicates step number. Very important is the touching face (red transparent). This touching face must be triangulated in the same way from both prisms. This is guaranteed with an arbitrary but prescribed point order.

Instead of constructing prisms seperately over each 3-simplex which would yield to a loop over all 3-simplices, each point is treated once and prisms hinging on this point are added to the triangulation. The loop over all 3-simplices is replaced with a loop over all points of the hypersurface. With the detailed description from section 2.3.1 it becomes apparent, that this method can be directly implemented to the Time-Evolution scheme. In the end it does not matter which of both prism methods are applied. Both yield the same results and the selection of one method is left to the author of the code.

Special Structured Triangulations for Closed and Open Spacetimes

Gentle and Miller described in [GM97] a triangulation for the closed universe which is not based on a triangulation of a cube. A closed universe is a four dimensional domain $T^3 \times \mathbb{R}$ consisting of a 3-torus T^3 and the timelike axis. With the results of Gentle and Miller, I was able to modify this approach to get structured triangulations for $B^3_R \times \mathbb{R}$ and $(B_R \backslash B_a) \times \mathbb{R}$ as well. This is meant in a topological sense. Here, B_R is a 3-ball of radius R. To triangulate a ball it is sufficient to triangulate a cube and to project vertices of the boundary onto a sphere of radius R. For finer grid resolutions, we can take a supercube. This is a triangulation of $n \times n \times n$ single cubes, stitched together, so they fill a huge cube completely with simplices. In this section, the projection of boundary points onto a sphere is not mentioned anymore.

Let us start with the original scheme from Gentle and Miller. To describe the decomposition of T^3 reasonably, integer point coordinates are used. Then T^3 is decomposed in the following way:

(I) A point set $\mathcal{P}_1 = \{2i, 2j, 2k\}$ is introduced with $i, j, k \in \mathbb{Z}$.

(II) A point set $\mathcal{P}_2 = \{2i+1, 2j+1, 2k+1\}$ is introduced with $i, j, k \in \mathbb{Z}$.

(III) Each point $\mathbf{p} \in \mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ is connected to six points $\mathbf{q}$ with $\mathbf{p} - \mathbf{q} = (2, 0, 0), (2, 0, 0), (0, 2, 0); (0, -2, 0), (0, 0, 2), (0, 0, -2)$.

(IV) Each point $\mathbf{p} \in \mathcal{P}$ is connected to eight points $\mathbf{q}$ with $\mathbf{p} - \mathbf{q} = (1, 1, 1)$, $(1, 1, -1)$, $(1, -1, 1)$, $(1, -1, -1)$, $(-1, 1, 1)$, $(-1, 1, -1)$, $(-1, -1, 1)$, $(-1, -1, -1)$

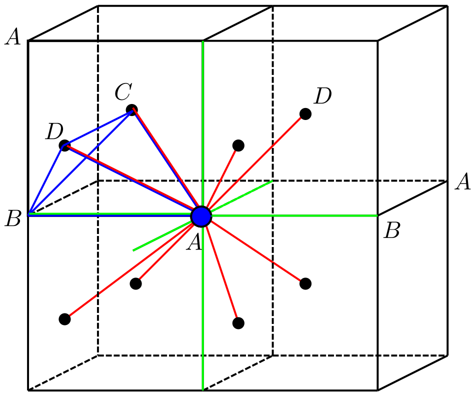


Fig. 3.11: Triangulation of T^3. 14 edges are emanating(green: 6 to B-points; red: 8 to C/D-points). One resulting simplex is highlighted in blue. Every simplex has one point from each point class.

Fig.(3.11) shows the structure of the triangulation. The edges described in point III in the algorithm above are colored green. Edges of point IV are colored red. To each one of the six edges of point III belong 4 simplices, connecting four points from each point class. That means, if $s = \{p_1, p_2, p_3, p_4\}$ denotes the simplex constructed with the four points p_1, p_2, p_3 and p_4, then $p_1 \in$ Point Class A, $p_2 \in$ Point Class B, $p_3 \in$ Point Class C and $p_4 \in$ Point Class D.

Now, we have an infinite triangulation of whole $\mathbb{R}^3$. By identifying opposite points we get a triangulation of the 3-torus T^3. The infinite index set $\mathbb{Z}$ reduces to a finite set. For example, if we are interested in a triangulation consisting of $3 \times 3 \times 3$ cubes, then the corresponding integer coordinates of points of point class A and B range through $\{0, 2, 4, 6\}$ and the integer coordinates of points of point class C and D range through $\{1, 3, 5, 7\}$. The integer point coordinate 8 is identified with 0 for A- and B-points, while the integer coordinate 7 is identified with 1 for C- and D-points. In this way we have modelled a closed space without any boundary.

If we are interested in a triangulation of $\Omega \subset \mathbb{R}^3$, we must not identify opposite points but we restrict the index set $\mathbb{Z}$ to a finite set only. But then, bumps occur at regular distance. For example, we are not able to model a triangulation of $3 \times 3 \times 3$ cubes with this method. To regain flat surfaces, that means to eliminate the bumps, additional technical effort is needed. Considering the example from above we must project C- and D-points which have integer coordinate 7 to the integer coordinate 6. But then simplices with zero volume occur. They have to be deleted from the triangulation. In this way, we get the desired triangulation but now we have also broken symmetries of the triangulation. For instance, the 3-diagonals do not have the same coordinate distance anymore when we look at points connected to the boundary with an edge.

If we are interested in an excised black hole domain, that is topologically $B_R \backslash B_a$, we can also model this with a modified version from the method described by GENTLE and MILLER. For example, it is possible to start with a $5 \times 5 \times 5$-triangulation without identifying points. Then we omit simplices which are completely part of the cube in the middle of the triangulation. The cube in the middle models the excised ball B_a. Now, some simplices still reach into this excised cube, so we have to project the relevant C- and D-points onto the inner boundary, that means we project these points onto the excised cube. This works only for C- and D-points which are far enough from edges and vertices of the excised cube. That is the case when the coordinate distance of the integer coordinates is greater than one for any component. If the C- and D-points are close to an edge of the excised cube then these points have to be doubled. The reason is, that now two faces of the excised cube meet at this edge. So, the point inside the excised cube has to be projected onto both faces. For C- and D-points near to a vertex of the excised cube three faces meet at this vertex. So, we have to triple these points and project each of them onto these three faces.

Now, we discussed structured three-dimensional triangulations for different topologies, starting from the triangulation idea of GENTLE and MILLER. To get a four-dimensional triangulation of $T^3 \times \mathbb{R}$ the prism method from section 3.2.2 is applied. The points are ordered by introducing four different point classes. Vividly, this is equivalent to a checkerboard structure on each of both point sets $\mathcal{P}_1$ and $\mathcal{P}_2$. In formulas the four point classes are defined by:

(A) $\mathbf{p} \in \mathcal{P}_1$ with $i + j + k =$even

(B) $\mathbf{p} \in \mathcal{P}_1$ with $i + j + k =$odd

(C) $\mathbf{p} \in \mathcal{P}_2$ with $i + j + k =$even

(D) $\mathbf{p} \in \mathcal{P}_2$ with $i + j + k =$odd

So, the point order can be summarized with $A \rightarrow B \rightarrow C \rightarrow D$. Within one point class it does not matter in which order the points are evolved. Also, each point of one point class can be evolved in parallel.

3.2.3 Usability of Triangulations for The Time-Evolution Scheme and Causality

As already shown, not every triangulation can be used for the Time-Evolution scheme. Two more requisites are necessary:

(1) The space between two hypersurfaces must be completely paved with simplices. No holes must occur. $\Rightarrow$ No inner simplices are allowed when applying triangulations of the 4-cube.

(2) The edges belonging to new equations must be determined.

In the case of Mara's triangulation shown in section 3.2.1, point 1 of the list above fails. Maybe there exists another Time-Evolution scheme for this kind of triangulation. But the ordinary one used in this work fails. Contrary, any triangulation which one gets by applying the prism method introduced in section 3.2.2 is valid. The prism method guarantees point one, there can not occur holes by construction. Each prism built ahead of its lower dimensional triangulation S completely fills $S \times [0,1]$.

For the discussion of point 2, it is sufficient to have a look at the prism method only. It includes all relevant cases. Fig.(3.12) shows the appearance of edge-unknown pairs in the prism method.

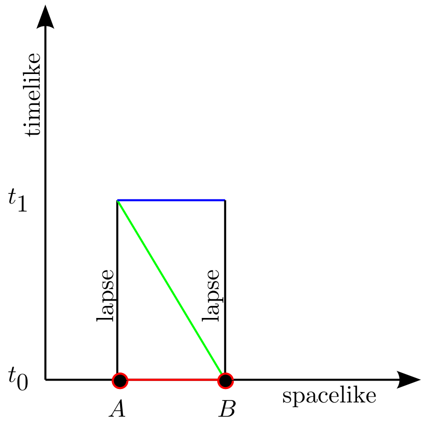


Fig. 3.12: With the prism method, always pairs of (new equation/new unknown) arise: (red/green) for point A and (green/blue) for point B.

No matter how complicated or unstructured the underlying three-dimensional hypersurface grid is, when we apply the prism method then there is a structured grid in timelike direction. Given any edge $\overline{AB}$ of T^3, the triangulation looks like Fig.(3.12) in timelike direction. Furthermore there are always pairs of edges which correspond to each other. One is the new unknown and one is the new edge. In the figure it is assumed that point A is evolved before point B. Then, at point A the red edge gives a new equation and the green edge is the new unknown. At point B the green edge gives a new equation and the blue edge is the new unknown. This shows also that the grid construction from Gentle and Miller shown in section 3.2.2 is valid. It applies the prism method in timelike direction too.

So, the number of unknowns and the number of equations match up in the prism method for each single point to be evolved before fixing lapse and shift edges. Lapse and shift is chosen according to section 3.1.4. Let us consider the Time-Evolution scheme applied to point P and let P^+ denote the point at the next hypersurface with spatial coordinates of P. All edges emanating from point P^+ are unknowns excluding the shift edges. Then, the edges which correspond to new equations can be determined via the equation/unknown-pair relationship. To get the number of unknowns or the number of new equations respectively, we can make use of a property of the prism method. Given the triangulation of the spatial hypersurface, we have to count the edges emanating from point P only and add one corresponding to the timelike edge. For instance, in the triangulation of Gentle and Miller there are $6+8=14$ edges (see section 3.2.2) in the hypersurface and so a total of 15 new equations per point arise.

When we look at arbitrary unstructured 3d grids evolved with the prism method, the number of equations is variable but always high enough to construct at least one Regge equation. This can be shown by the following example. Let us consider an example in $2+1$ dimensions, sketched in figure Fig.(3.13). In section 3.1.4 it was shown that lapse and shift have to be

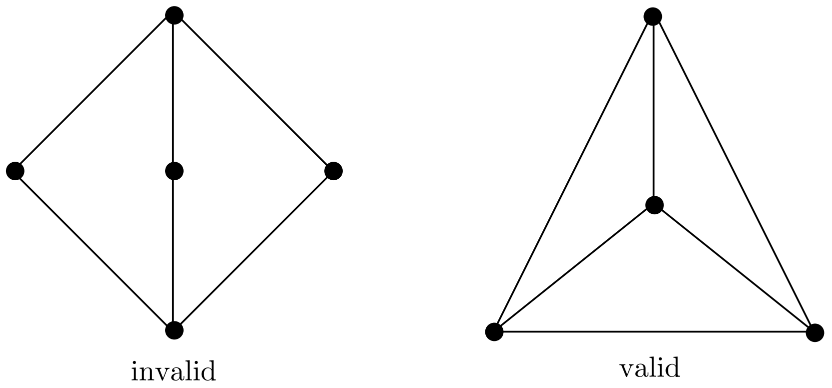


Fig. 3.13: *Left.* It is possible to surround a point with two triangles but then the triangulation is invalid. *Right.* Three is the minimum number of triangles to surround a point in a valid triangulation.

fixed. In $2 + 1$ dimensions three edges must be fixed before any Regge equation can be calculated. Given an arbitrary evolution point the two shift edges for this point are part of the hypersurface. We can restrict the problem to the hypersurface. If we are able to construct a two-dimensional grid where one point is connected with only two others, then after fixing the necessary two shift edges there would be no unknown left. That would mean, that Regge Calculus applied to unstructured grids could fail principally (aside from many other issues like convergence for instance). Fortunately, this is impossible. Since at least three triangles are necessary in a valid triangulation to surround one point, at least three edges emanate from this point. This can be generalized to three dimensions and four dimensions as well. For example, in three dimensions, it is impossible to surround one point with less than four tetrahedrons in a valid triangulation. The result of the consideration is: If the four-dimensional domain is triangulated using the prism method applied to an arbitrary 3d grid, then there exists at least one unknown squared edge length to be calculated via Regge equations.

Causality and Courant condition

A last aspect to discuss is the stepsize of time. Let P denote the point to be evolved and P^+ denote its pendant one timestep further. The neighbour point is noted with A or A^+ respectively. A natural restriction comes from causality: The past light cone of P^+ must not contain the neighbor point A otherwise a region of the hypersurface would influence P^+ theoretically which is not supposed to influence P^+ by means of the constructed local Time-Evolution scheme. Analogous, the past light cone of A^+ must not contain P.

In Fig.(**??**) this situation is sketched for both cases. Via the left hand side a $1 : 1$ relation between the length of the minimal spacelike edge and timelike edge can be detected. The right hand side points out that edges can occur which cross both hypersurfaces. These edges are spacelike too. Due to the Minkowski metric the edge length is smaller if an edge crosses two hypersurfaces. Moreover, a ratio of $1 : \sqrt{2}$ must be demanded at least to guarantee that the past light cone of A^+ does not have point P in common. This picture holds true for any three-dimensional triangulation with an applied prism method.

If we specify the used triangulation to the triangulation of Gentle and Miller we can derive a Courant ratio theoretically by demanding causality. There, the spatial discretization width Δx is the distance of points along an axis. But due to the body centred cubic structure of the triangulation also diagonals arise which have a smaller distance than Δx. Regarding the construction scheme from section 3.2.2 Δx corresponds to the separation vector $(200)^T$. Then the diagonal separation vector $(111)^T$ corresponds to $\Delta x \cdot \sqrt{3}/2$ since the usual norm in Minkowski metric of $||(200)^T|| = 2$ and $||(111)^T|| = \sqrt{3}$. If we consider now such a diagonal in the hypersurface there exists also a corresponding diagonal which crosses two hypersurfaces. Then the corresponding norm is: $||(1111)^T|| = \sqrt{2}$. So, if Δx describes the spatial discretization width in axis-direction, then $\Delta x/\sqrt{2}$ is the minimum spacelike edge length in the four-dimensional triangulation. Regarding this the following Courant ratio is expected for GENTLE-MILLER triangulations if causality is necessary *and* sufficient:

$$\frac{\Delta t}{\text{minimum spacelike edge length}} = \frac{1}{\sqrt{2}} = \frac{\Delta t}{\Delta x/\sqrt{2}} \tag{3.13}$$

$$\Rightarrow \frac{\Delta t}{\Delta x} = \frac{1}{2}\,. \tag{3.14}$$

In [GM97] much stronger ratios were applied. For the Kasner universe, they applied 1 : 5 for the initial hypersurface. For the flat Kasner universe, they used a ratio from 2 : 5.

3.3 Basic Algorithms Used in the Implementation of The Regge equations

3.3.1 General Layout of the Code and the Data Structure

The code applied to several problems in this thesis was written from scratch and was developed completely by my own. The code was written in the programming language `C++`. It uses the containers `map`, `vector` and `list` from the Standard Template Library (`STL`). For solving general linear systems of equations, extern routines from the `LAPACK` were used. Three different containers were assigned to three different parts of the triangulation, namely it is sufficient to treat points, edges and simplices. These containers give information on each geometrical object and they refer to each other according to the triangulation. Fig.(3.14) shows this construct.

First, the left hand side of the figure is discussed. Simplices, points and edges are connected with pointers in a natural manner. The 4-simplex refers to its five points and ten edges. The edges refer to its two points and to the simplices they are part of. The latter number is variable and changes from edge to edge. The simplex data structure does not hold any data but works as an interface structure only.

The points hold technical information which are important to the Time-Evolution scheme. For structured grids, the location of the point is described with an integer vector. This 4-tuple of integer stores topological information which is needed to detect neighbor points for instance. This is necessary to search for the new unknowns and new equations, as well as to determine lapse and shift edges. Furthermore, the initial coordinates which are only known for the initial slice are stored. From the first timestep on it is unclear which coordinates the points have. But by assuming only small changes, it can be used to detect errors. Only if the evolved point and the point described with the analytical coordinates do not differ a lot in location, a meaningful error measure can be calculated. The stored data of a point

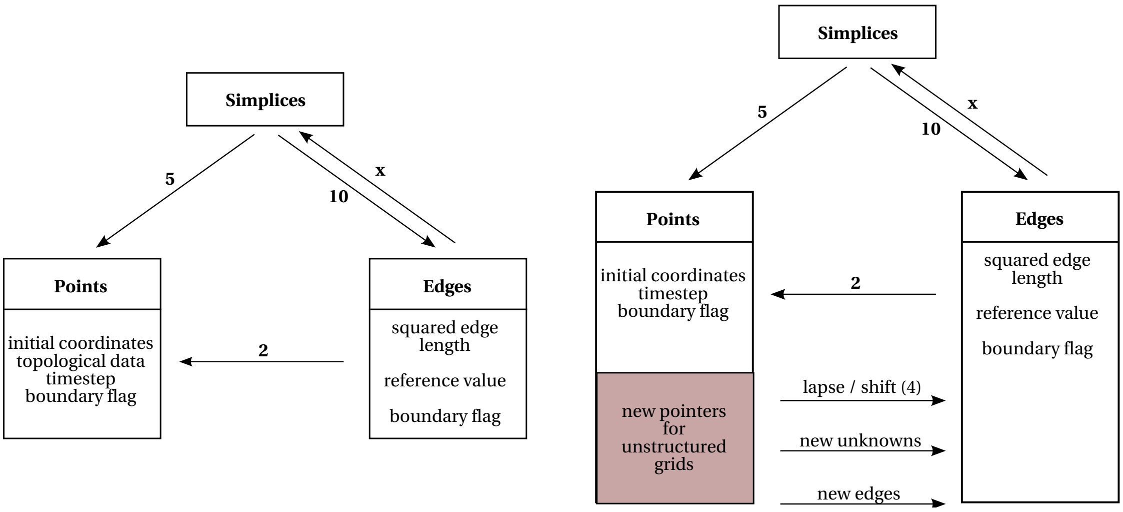


Fig. 3.14: *Left.* Data structure for structured grids. *Right.* Data structure for unstructured grids.

also contains a boundary flag, describing if the point is part of the boundary of the spatial domain or if it is inside.

The datastructure for the edge stores the squared edge length and its reference value. The reference value is calculated from the analytical point coordinates and therefore is known exactly for the initial data only. The edge datastructure stores also a boundary flag, describing if a Regge equation corresponds to this edge. With this data structure a complexity of $\mathcal{O}(N \log N)$ is guaranteed. The $\log N$ term comes from search routines on global containers inside a global loop which were realized with the container class `map` of the `STL`. Such a search routine appears for instance when an edge corresponding to two points is searched in a global edge map.

An improved data structure for unstructured grids is shown at the right hand side of Fig.(3.14). For unstructured grids it is not possible to store a whole integer vector. Only the remaining structured grid in timelike direction is stored with the `timestep` value. Due to the prism method it is possible to set up additional pointers which replaces the job of the above mentioned search routines. Therefore, this approach has a complexity of $\mathcal{O}(N)$ only but can be less flexible when applied to structured grids. Therefore both approaches were employed.

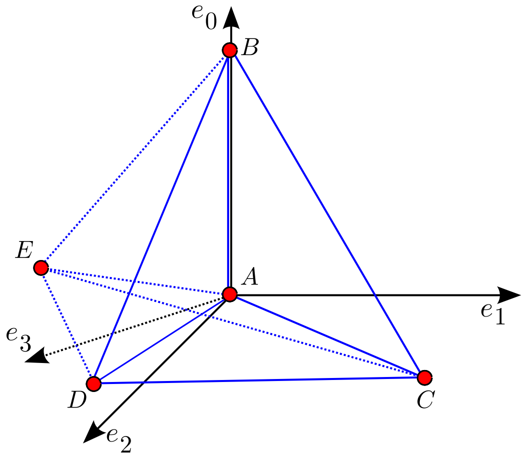
3.3.2 Calculation of the Deficit Angle

The *deficit angle* is the paramount quantity in Regge Calculus. It can be calculated by the edge lengths only. That is how the equations are constructed but it is not trivial. The next two sections will go into detail how the deficit angle is calculated in the code. The method was devolped by my own. It is different from the method described by Sorkin in [Sor74]. There, the vector is parallel transported through simplices and then the deficit angle is calculated. But here, each dihedral angle between simplex faces are calculated separately. The implemented method follows slightly the ideas found in [MTW73]. Unfortunately all other papers take this routine for granted, so no further comparison could be made.

First, an orthogonal coordinate system is constructed in each simplex, then the normal vectors are calculated on both three-dimensional faces and finally the definition of the cosine gives rise to an equality of the dihedral angle and a complex inverse cosine of an argument which itself is complex. Fortunately all values are either pure real or pure imaginary which simplifies the problem significantly.

The Local Orthogonal Coordinate System

Fig.(3.15) shows a 4-simplex in a two-dimensional projection. Given the ten edge lengths of a simplex defined by the points A, B, C, D and E and given the bone triangle $\triangle ABC$ the aim is to construct a basis $\{\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ which is normal and orthogonal. That means its metric is demanded to be the flat Minkowski metric in its standard form. In cartesian coordinates, the flat metric is $\eta_{\mu\nu} = \text{diag}\,\eta_{\mu\nu} = (-1, 1, 1, 1)$. Unfortunately the place where the sign occurs is not known a priori but, as an advantage compared to the Sorkin scheme, it is detected directly with the algorithm.



vector	symbol	edge	symbol
$\overrightarrow{AB}$	$\mathbf{a}$	$\overline{AB}$	l_a
$\overrightarrow{AC}$	$\mathbf{b}$	$\overline{AC}$	l_b
$\overrightarrow{AD}$	$\mathbf{c}$	$\overline{AD}$	l_c
$\overrightarrow{AE}$	$\mathbf{d}$	$\overline{AE}$	l_d

Fig. 3.15: *Left.* A 4-simplex in an orthonormal coordinate system $\{\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. *Right.* Name conventions in the derivation.

The vector $\overrightarrow{OA}$ points to the origin of the local coordinate system. Then the vector $\overrightarrow{AB}$ is expressed by

$$\overrightarrow{AB} := \mathbf{a} = a_0 \mathbf{e}_0 \,. \tag{3.15}$$

Let s_i, $i = 0, 1, 2, 3$, denote the sign of the standard Minkowski metric. That means it expresses at which component the minus sign occurs. Further let ${l_a}^2$ denote the squared edge length of the edge $\overline{AB}$. Then from

$$\langle \mathbf{a}, \mathbf{a} \rangle = s_0 {a_0}^2 = {l_a}^2 \tag{3.16}$$

we get the coordinate a_0. The sign s_0 we get from

$$a_0 = \sqrt{|{l_a}^2|} \quad \text{and} \quad s_0 = \text{sign}\,({l_a}^2) \,.$$

Finally, with Eq.(3.15) we get also an expression for the first basis vector,

$$\mathbf{e}_0 = \frac{\mathbf{a}}{a_0} \,.$$

In the following, let be $\overrightarrow{AC} = \mathbf{b}$, $\overrightarrow{AD} = \mathbf{c}$ and $\overrightarrow{AE} = \mathbf{d}$, as well as $\overline{AC}^2 = {l_b}^2$, $\overline{AD}^2 = {l_c}^2$ and $\overline{AE}^2 = {l_d}^2$. The next vectors to be constructed can be expressed as

$$\mathbf{b} = b_0\mathbf{e}_0 + b_1\mathbf{e}_1\,, \tag{3.17}$$

$$\mathbf{c} = c_0\mathbf{e}_0 + c_1\mathbf{e}_1 + c_2\mathbf{e}_2\,, \tag{3.18}$$

$$\mathbf{d} = d_0\mathbf{e}_0 + d_1\mathbf{e}_1 + d_2\mathbf{e}_2 + d_3\mathbf{e}_3\,. \tag{3.19}$$

Now, the calculation of the first component of $\mathbf{b}$ yields

$$\begin{aligned} s_0 b_0 = \langle \mathbf{b}, \mathbf{e}_0 \rangle = \langle \mathbf{b}, \frac{\mathbf{a}}{a_0} \rangle &= \frac{1}{a_0}\langle \mathbf{a}, \mathbf{b} \rangle\,. \\ \Longrightarrow \quad b_0 &= \frac{s_0}{a_0}\langle \mathbf{a}, \mathbf{b} \rangle\,. \end{aligned} \tag{3.20}$$

At this place a formula for the scalar product of two vectors which consists only of squared edge lengths is needed. By using the cosine theorem we get a general equality for a triangle $\triangle UVW$ and two vectors $\overrightarrow{UV} = \mathbf{v}$, $\overrightarrow{UW} = \mathbf{w}$,

$$\begin{aligned} \langle \mathbf{v}, \mathbf{w} \rangle :=& \cos\gamma ||\mathbf{v}||\,||\mathbf{w}|| = \frac{\overline{UV}^2 + \overline{UW}^2 - \overline{VW}^2}{2||\mathbf{v}||\,||\mathbf{w}||} \cdot ||\mathbf{v}|| \cdot ||\mathbf{w}|| \\ =& \frac{1}{2}\left(\overline{UV}^2 + \overline{UW}^2 - \overline{VW}^2\right)\,. \end{aligned}$$

With this formula every scalar product can be calculated easily. The components c_0 and d_0 can be computed directly from Eq.(3.20) just by exchanging symbols $b \leftrightarrow c$, $b \leftrightarrow d$. To get the next basis vector and the last component of $\mathbf{b}$ we apply the idea of Eq.(3.16):

$$\langle \mathbf{b}, \mathbf{b} \rangle = s_0{b_0}^2 + s_1{b_1}^2 = {l_b}^2 \tag{3.21}$$

$$\Longrightarrow \quad b_1 = \sqrt{|{l_b}^2 - s_0{b_0}^2|}\,, \tag{3.22}$$

$$s_1 = \mathrm{sign}\,({l_b}^2 - s_0{b_0}^2)\,. \tag{3.23}$$

$$\text{With Eq.(3.17)} \quad \Longrightarrow \quad \mathbf{e}_1 = \frac{1}{b_1}\left(\mathbf{b} - \frac{b_0}{a_0}\mathbf{a}\right)\,. \tag{3.24}$$

With the new basis vector $\mathbf{e}_1$ the components c_1 and d_1 are easy to calculate with the scalar products $\langle \mathbf{c}, \mathbf{e}_1 \rangle$ and $\langle \mathbf{d}, \mathbf{e}_1 \rangle$. We have only to replace the basis vector $\mathbf{e}_1$ with the derived equation Eq.(3.24). Now, c_2 and s_2 are calculated in the same manner as b_1 and s_1 in Eq.(3.22) and Eq.(3.23):

$$\begin{aligned} \langle \mathbf{c}, \mathbf{c} \rangle &= s_0{c_0}^2 + s_1{c_1}^2 + s_2{c_2}^2 = {l_c}^2 \\ \Longrightarrow \quad c_2 &= \sqrt{|{l_c}^2 - s_0{c_0}^2 - s_1{c_1}^2|} \\ \Longrightarrow \quad s_2 &= \mathrm{sign}\,({l_c}^2 - s_0{c_0}^2 - s_1{c_1}^2)\,. \end{aligned}$$

The last necessary basis vector becomes with Eq.(3.18),

$$\mathbf{e}_2 = \frac{1}{c_2}\left(\langle \mathbf{c},\mathbf{d}\rangle + \frac{1}{a_0}\left(\frac{c_1 b_0}{b_1} - c_0\right)\langle \mathbf{a},\mathbf{d}\rangle - \frac{c_1}{b_1}\langle \mathbf{b},\mathbf{d}\rangle\right).$$

This gives d_2 by calculating $\langle \mathbf{d},\mathbf{e}_2\rangle$. Then d_3 and s_3 are determined again analogous to b_1 and s_1 in Eq.(3.22) and Eq.(3.23):

$$\begin{aligned} \langle \mathbf{d},\mathbf{d}\rangle &= s_0 {d_0}^2 + s_1 {d_1}^2 + s_2 {d_2}^2 + s_3 {d_3}^2 = {l_d}^2 \\ \implies \quad d_3 &= \sqrt{|{l_d}^2 - s_0 {d_0}^2 - s_1 {d_1}^2 - s_2 {d_2}^2|} \\ \implies \quad s_3 &= \operatorname{sign}\left({l_d}^2 - s_0 {d_0}^2 - s_1 {d_1}^2 - s_2 {d_2}^2\right). \end{aligned}$$

Now, the basis is complete and the signature of the metric has been detected. An explicit formula of $\mathbf{e}_3$ is not used but could be derived if necessary. The detected metric $\eta_{\mu\nu} = \operatorname{diag}(s_0, s_1, s_2, s_3)$ can be used to reveal errors in the code. If the metric has more or less than one minus sign than the code is crashed already or the start values for the Newton scheme are chosen in a wrong way.

Normal Vectors and the Formula of the Dihedral Angle of a Simplex

From the previous section we got four vectors $\mathbf{a} = (a_0, 0, 0, 0)^T$, $\mathbf{b} = (b_0, b_1, 0, 0)^T$, $\mathbf{c} = (c_0, c_1, c_2, 0)^T$ and $\mathbf{d} = (d_0, d_1, d_2, d_3)^T$. These span the 4-simplex $ABCDE$ in a standard Minkowski metric. By assumption, that the bone is $\triangle ABC$ the two faces for the dihedral angle are $ABCD$ and $ABCE$. Using three arbitrary vectors $\mathbf{u}, \mathbf{v}$ and $\mathbf{w}$ the normal vector in any metric can be expressed in component notation as

$$n^\alpha = \varepsilon^\alpha{}_{\beta\gamma\delta}\, u^\beta\, v^\gamma\, w^\delta\,. \tag{3.25}$$

Here, the Einstein summation convention holds and all indices are running from 0 to 3. ε is the *Levi Civita Symbol* and is defined as

$$\varepsilon_{\alpha\beta\gamma\delta} := \begin{cases} 1 & \text{if } [\alpha\beta\gamma\delta] \text{ is an even permutation,} \\ -1 & \text{if } [\alpha\beta\gamma\delta] \text{ is an odd permutation,} \\ 0 & \text{else}\,. \end{cases}$$

Before we apply this definition we have to draw an index. Also, the metric is already fixed to the standard Minkowski metric $\eta_{\mu\nu}$:

$$\begin{aligned} n^\alpha &= \eta^{\alpha\lambda}\, \varepsilon_{\lambda\beta\gamma\delta}\, u^\beta\, v^\gamma\, w^\delta \\ &= s_\alpha\, \varepsilon_{\alpha\beta\gamma\delta}\, u^\beta\, v^\gamma\, w^\delta\,. \end{aligned}$$

The s_α is the sign at position α of the standard Minkowski metric and was introduced in the previous section. Let $\mathbf{n}_1$ denote the normal vector on $ABCD$ and let $\mathbf{n}_2$ be the normal vector on $ABCE$. Because the simplex-spanning vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\mathbf{d}$ have a lot of zero components, the normal vectors have many zero components too. A small calculation gives us:

$$\mathbf{n}_1 = (0,0,0,{n_1}^3) \qquad \text{and}$$
$$\mathbf{n}_2 = (0,0,{n_2}^2,{n_2}^3)\,.$$

Then, the components are:

$$\begin{aligned}
{n_1}^3 &= -s_3\, a^0\, b^1\, c^2\,,\\
{n_2}^2 &= s_2\, a^0\, b^1\, d^3\,,\\
{n_2}^3 &= -s_3\, a^0\, b^1\, d^2\,.
\end{aligned}$$

At last a statement concerning the orientation of the normal vectors is necessary. The ordinary definition of the cosine gives the correct dihedral angle if span $\{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{n}_1\}$ and span $\{\mathbf{a},\mathbf{b},\mathbf{d},\mathbf{n}_2\}$ have the same orientation. This is guaranteed by the definition of the normal vector in Eq.(3.25). So, the dihedral angle α between the faces $ABCD$ and $ABCE$ is calculated with

$$\begin{aligned}
\cos\alpha &= \frac{\langle \mathbf{n}_1,\mathbf{n}_2\rangle}{\sqrt{\langle \mathbf{n}_1,\mathbf{n}_1\rangle \cdot \langle \mathbf{n}_2,\mathbf{n}_2\rangle}}\,.\\
&= \frac{s_3 \cdot {n_1}^3 {n_2}^3}{\sqrt{s_3 {n_1}^3 {n_1}^3 \cdot (s_2 {n_2}^2 {n_2}^2 + s_3 {n_2}^3 {n_2}^3)}}\\
&= \frac{s_3 d^2}{\sqrt{s_2 s_3 (d^3)^2 + (d^2)^2}}\,.
\end{aligned} \tag{3.26}$$

We see that in the case of a timelike bone ($s_0 = -1$ or $s_1 = -1$), the right hand side is in the interval $[-1,1]$. Therefore the angle α is real valued as expected because a timelike bone corresponds to a spacelike image space $\mathcal{M}$ in which all angles are Euclidean.

In the case of a spacelike bone ($s_2 = -1$ or $s_3 = -1$) we see that the root argument becomes negative if $d^3 > d^2$ and the right hand side is purely imaginary. Or, if $d^3 < d^2$ the root argument is positive but the numerator is now greater than the denominator. This yields a real value not in the interval $[-1,1]$. These three different cases will be discussed in the next section.

The Inverse Cosine

The derivation of the deficit angle confronts one with the inverse cosine in Eq.(3.26). This is a multivalued map, so one complex value is mapped to an infinity amount of complex values in general.

$$\arccos : \mathbb{C} \to \mathbb{C}\,, \tag{3.27}$$

$$\arccos(z) := \frac{1}{2}\pi + i\,\ln(iz + \sqrt{1-z^2})\,. \tag{3.28}$$

The first step is to restrict this map to a singlevalued map to get an invertible function. In this case the output values are *principal values.* This brings the problem of branch cuts, that are discontinuities in the map due to the cutoff of all other output values. Usually this is done at the intervals $[-\infty,-1]$ and $[1,\infty]$. For calculating deficit angles the argument of the inverse cosine function can be either pure real or pure imaginary. That is because Eq.(3.26) can have square roots of negative reals. So we can specify the domain further to $\arccos : \mathbb{X} := \mathbb{R} \cup i\mathbb{R} \Rightarrow \mathbb{C}$.

Now, the output values of this new domain show constant real parts except for the interval $[-1,1]$. In the latter case the imaginary part is zero of course. The constant real parts are not of interest because they sum up to zero later in the calculation of the deficit angle. So we can summarize, that for spacelike bones the imaginary part of the dihedral angle is important and for timelike bones the real part is important. So the Inverse Cosine decomposes to three maps of interest which can be plotted. Also, simplified formulas can be derived starting from Eq.(3.27):

$$\begin{aligned}
f_1(x):\quad & [-1,1] \rightarrow [0,\pi] & \Longleftrightarrow \qquad f_1(x) &= \mathrm{Re}\left(\arccos(x)\right), \\
& & &= \mathrm{acos}(x), \\
f_2(x):\quad & [-\infty,-1]\cup[1,\infty] \rightarrow \mathbb{R} & \Longleftrightarrow \qquad f_2(x) &= \mathrm{Im}\left(\arccos(x)\right) \\
& & &= \ln(|x+\sqrt{x^2-1}|), \\
f_3(x):\quad & \mathbb{R} \rightarrow \mathbb{R} & \Longleftrightarrow \qquad f_3(x) &= \mathrm{Im}\left(\arccos(ix)\right) \\
& & &= \ln(|x-\sqrt{x^2+1}|).
\end{aligned}$$

This real valued formulas can be implemented directly in any programming language. For instance, in `C++` ordinary `acos`, `log` and `sqrt` functions from the C Standard Library can be used.

By specifying the inverse cosine in this way the branch cuts are out of scope. Using a naive approach could fail. For instance an argument $z = 2 \pm 10^{-15} i$ would be mapped to $\mathrm{Im}\left(\arccos(z)\right) \approx \mp 1.31\, i$ due to the discontinuouity.

3.3.3 The Composition of One Regge Equation

Having an algorithm for the calculation of a dihedral angle θ the Regge Equations are composed quite easily. Only local objects are needed, so it is feasible to use pointers. Given a function `double G(Edge e)` which maps an edge e to the value of the Regge equation r, first the simplices which are hinging on that edge are accessed by the appropriate pointer array:

$$\mathrm{Edge}\, e \quad \longrightarrow \quad \text{set of simplices}\, \{s_1, s_2, \ldots\}\,.$$

A loop over this local set of simplices is started and in each loop all possible bones b containing the given edge are constructed. Then, the dihedral angle $\theta_{b,s}$ of this bone (subscript b) in the actual simplex (subscript s) is calculated and stored in an array `M` mapping the bone $\theta_{b,s}$ to the negative sum of the already calculated dihedral angles. When the simplex loop is finished, 2π is added to the real valued angles. As the result, the map `M` consists of the deficit angles.

At last the area of the bone is calculated and the terms of kind

$$\frac{A(b)}{M(b)} \cdot ({l_e}^2 - {l_1}^2 - {l_2}^2)\,, \tag{3.29}$$

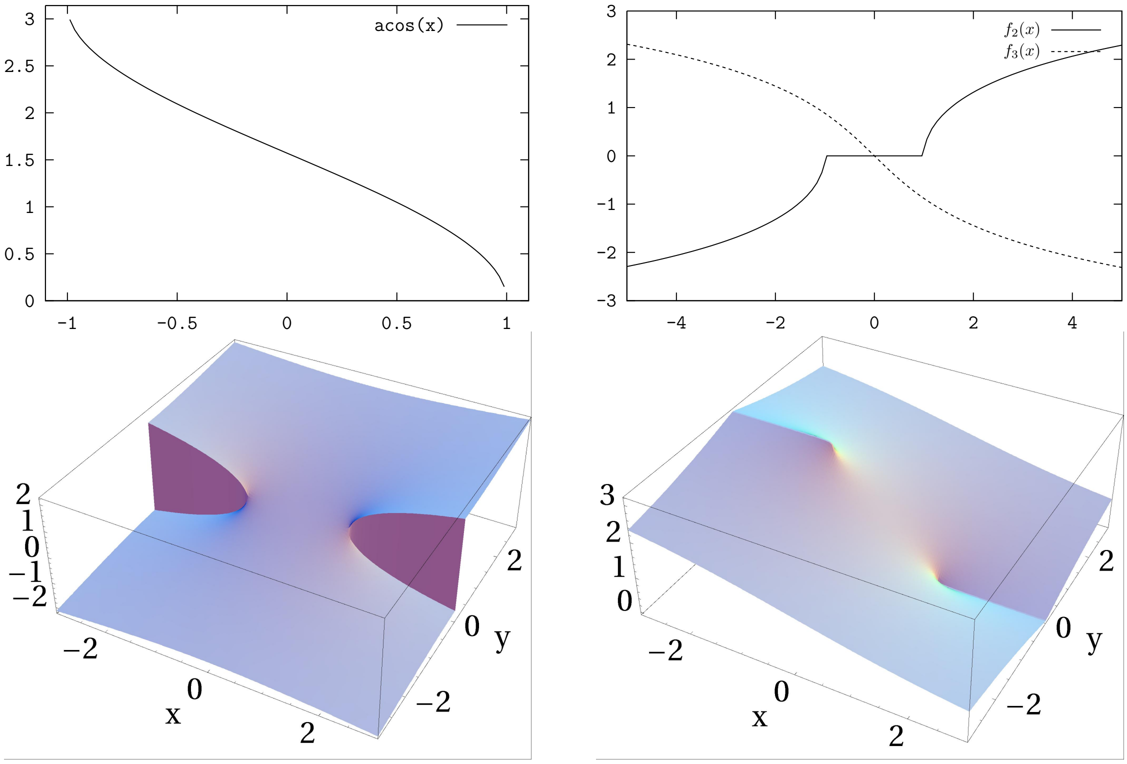


Fig. 3.16: *Top Left:* $f_1(x)$. Ordinary inverse cosine. *Top Right:* $f_2(x)$. Imaginary part of inverse cosine of real axis except the interval $[-1, 1]$. $f_3(x)$: Imaginary part of inverse cosine of imaginary axis. *Bottom Left:* Imaginary part of inverse cosine. The Branch cuts can be seen. *Bottom Right:* Real part of inverse cosine.

are summed up which gives the value of the Regge equation. In Eq.(3.29) ${l_e}^2$ is the squared length of edge e and ${l_1}^2$ and ${l_2}^2$ are the squared edge lengths of the two other edges of the bone.

This routine guarantees local effort and therefore can be expressed as a constant value in the global effort. The routine is summarized in algorithm 4.

3.3.4 The Time-Evolution Algorithm

Given a three-dimensional decomposition of the space-domain the algorithm 5 is used for Time-Evolution with Regge Calculus in the developed code. The following subsections will describe this algorithm in detail.

Setting Up the Problem

The idea of setting up the problem is sketched in Fig.(3.17).
In line `01` the three-dimensional triangulation is read. If we want to implement parallel algorithms or a special point order for evolution the initial triangulation has to provide information about point classes. As argued before, points of one point class are *not* connected with an edge. Such points can in principle be calculated in parallel. In line `04` the initial slice is evolved in timelike direction with the *prism method* described in section 3.2.2. This is done twice with the reason to keep this so-built four-dimensional structure static for the remaining algorithm. By this, we do not need to care about grid generation anymore. A small drawback in flexibility concerns refinement techniques. These can not directly be plugged in but with

Algorithm 4 Algorithm for composing a Regge equation.
Function G. *Input value*: Edge e. *Output value*: Value of Regge equation r.

```
M=0
Get simplexes $s_i$, $i = 1, 2, \dots, N_s$ with $e \subset s_i$ {Access by pointer.}
for all $s_i$, $i = 1, 2, \dots$ do
    Construct the three bones $b_j$, $j = 1, 2, 3$ of simplex $s_i$ with $e \subset b_j$
    for all $b_j$, $j = 1, 2, 3$ do
        Calculate dihedral angle $\theta_{b,s}$ {Input values: Array of ten edge lengths of simplex $s_i$
        ordered with respect to bone $b_j$.}
        $M(b_j) = M(b_j) - \theta_{b,s}$ {Sum up dihedral angles belonging to bone $b_j$ in array $M$.}
    end for
end for
Add $2\pi$ to angles belonging to timelike bones in array $M$
$r = 0$
for all $b_i$, $i = 1, 2, \dots$ with $e \subset b_i$ do
    Calculate area of bone $A(b_i)$
    $v = {l_e}^2 - {l_1}^2 - {l_2}^2$
    $r = r + A(b_i)/M(b_i) \cdot v$
end for
return $r$
```

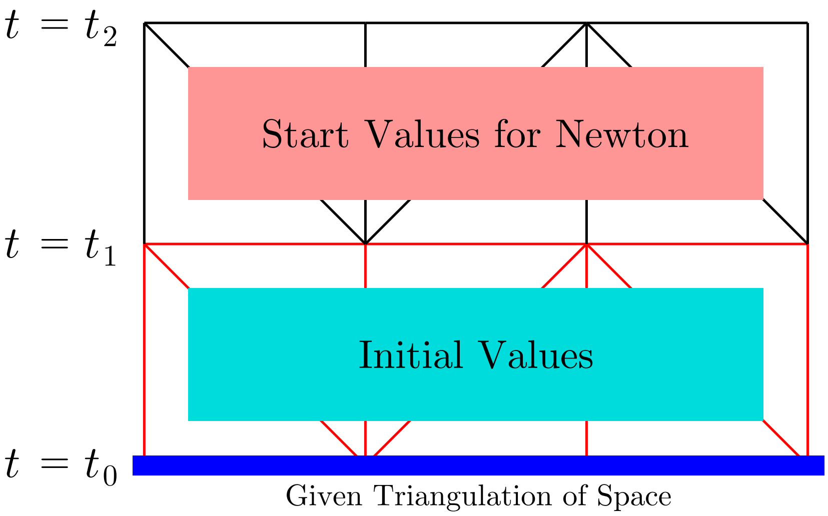


Fig. 3.17: Static triangulation. A given 3d triangulation S (blue) is evolved in time with prism method. $S \times [t_0, t_1]$ is needed for initial values. $S \times [t_1, t_2]$ is needed for newton start value and new data.

moderate effort this should be achieveable. As a last step (line `05`) to get ready for Time-Evolution, the initial values of the squared edge lengths on hypersurface S_0, S_1 and between both must be prescribed. The values between S_1 and S_2 and on slice S_2 are the start values for the Newton method. They need also to be set.

The Time-Evolution Loop

Each timestep starts with a loop over all point classes (line `09`). The algorithm inside this loop is parallelizeable. For sake of simplicity it is kept serial in this work. Line `10` loops over all points of the fixed point class i. Also, points on slice S_1 are evolved only because this is

Algorithm 5 Main Algorithm.

```
1:  Set up triangulation and data structure
2:  Read 3d triangulation for first slice $S_0$
3:  Assign point classes
4:  Evolve slice $S_0$ with prism method twice ($\rightarrow$ $S_1$, $S_2$) {From now on the triangulation is static.}
5:  Set initial edge lengths on $S_0$, $S_1$ and between both
6:  $N_t = 2$
7:  while $N_t <$ MAXITER do
8:      $N_t = N_t + 1$
9:      for all point classes $i = 1, 2, \dots$ do
10:         for all points $p \notin \partial\Omega$ from point class $i$ and $t = t_1$ do
11:             Determine $N_{\text{eq}}$ new equations {i.e. new inner edges.}
12:             Determine $N_{\text{var}}$ new edges
13:             Fix gauge freedom (i.e. fix squared edge length of 4 new edges) {Fix lapse edges and shift edges}
14:             while Error > TOL do
15:                 Ordinary Newton Step
16:                 if Error increased then
17:                     Apply Damping Scheme
18:                 end if
19:             end while
20:         end for
21:     end for
22:     Copy all data from hypersurface to the one below $S_2 \rightarrow S_1$, $S_1 \rightarrow S_0$. Handle triangulation between hypersurfaces analogous
23:     Prescribe squared edge lengths at boundary $\partial\Omega$ and prescribe squared edge lengths which belong schematically to points at boundary
24: end while
```

the uppermost physical time slice. In line `11` and `12` the new equations and the new edges *with respect to the actual point p* are determined. That means, if we would enhance the triangulation with the Time-Evolution scheme manually, we would naturally get these new edges and the new equations. At this point, the number of new equations and new edges are equal in all relevant cases presented in this work.

Before Regge equations can be solved the new evolved point must be fixed in geometry by prescribing four squared edge lengths (line `13`). Practically, this is achieved with stating four equations by which this four squared edge lengths can be calculated. The next step solves the system of non-linear equations with a Newton method using a QR method inside, if $N_{\text{eq}} \neq N_{\text{var}}$ and a LU method otherwise. Procedures of the `LAPACK` are used. If the Newton step is not successful, that means the error increases, than a damping scheme is applied.

In line `22` the new calculated squared edge lengths of slice S_1 and S_2 and between S_1 and S_2 are copied one timestep downwards. Physically, that means we move on to the next timestep. As a positive side effect the new Newton start values are already set: They are the old values from the former timestep.

3.3.5 Speedup Methods and Parallelization

Considering the presented algorithm there are lots of methods to speed up the code signficantly. In the following the investigated methods with the highest impact on reducing computational effort are presented. Let us have a close look at the Newton scheme. As derived before, in each Newton step a linear system of equation has to be solved. With the Jacobian matrix J, the solution vector $\mathbf{x}$ and the right-hand side vector $\mathbf{rhs}$ the system reads:

$$J\mathbf{x} = \mathbf{rhs}\,.$$

Given such a system, the optimized routines from `LAPACK` were used. Here, the chance to speed up the code is very low since these routines were developed for a long time with high manpower. On the other hand we can have a look of how to construct the matrix J. It is sufficient to consider a quadratic system to point out the speedup routines. Assuming a matrix dimension of $N = 11$, which would be the case for the structured grid from GENTLE and MILLER and applying the LU scheme, we have to determine 121 derivatives of Regge equations. Since the analytical calculation of the components takes a lot of time, is complex and therefore error-attracting, numerical derivatives are used. Ordinary finite differencing gives us derivatives of arbitrary accuracy. Let us consider the left-hand difference and the central-difference scheme.from Eq.(3.10) and Eq.(3.11). If we take the latter one, we have to calculate a Regge equation twice for each Jacobian matrix component. These are $G_i(l^2 + h)$ and $G_i(l^2 - h)$. Here G denotes the Regge equation. If we take the left-sided difference, we have to calculate only one Regge equation namely $G_i(l^2 + h)$ since the $G_i(l^2)$ are already avaiable via the right-hand side vector $\mathbf{rhs}$. When presuming numerical experiments give identical results for both approaches then the left-hand sided difference is preferable. This method of speedup is the first considered. The method is called *Left-hand difference* below.

The second speedup procedure comes from the fact that a lot of components of the Jacobian matrix are zero. They can be determined with the following idea. Since the matrix component J_{ij} describes the derivation of Regge equation G_i with respect to the squared edge length ${l_j}^2$ the matrix component is zero when the edges e_i and e_j are not connected with a simplex. This can be seen via the construction scheme of a Regge equation G_i. It comprises all simplices hinging at this edge e_i and not more. Only squared edge length from all simplices hinging at edge e_i do influence the calculation. Therefore, if e_i and e_j are not part of one and the same simplex, the matrix component J_{ij} is zero. The method is called *Zero Component Search* below.

Another method of speedup and the last considered here is to calculate the Jacobian J only once in the whole Newton scheme instead of calculating it again and again in each iteration step. This method is part of the so called Non-exact Newton methods. The speed of convergence drops to linear order. But in practice it turns out that in most cases the first iteration step of the Newton scheme is already very good. All steps afterwards do change the squared edge lengths in $\mathbf{x}$ slightly only. The method is called *Inexact Newton* below.

Anticipating the Schwarzschild metric from section 3.4.6 all three speedup methods were applied to this metric with $M = 1$ posed on a ball B_R of radius $R = 1000$. An unstructured 3d grid from the external grid generator `NETGEN` was used. There, the setting `very fine` was activated. By applying the prism method to evolve the 3d grid to a 4d grid as shown in algortithm 5, this gives us 389 inner nodes to evolve and 13930 edges in the whole 4d triangulation. Each of these edges are needed to calculate the deficit angle in the Time-Evolution scheme. The timestep was chosen to be $dt = 0,2$. The timestep fulfills a very strong Courant condition with the ratio of smallest spatial edge length to timestep size with 50 : 1. The Time-Evolution was run for 10 iteration steps corresponding to a time difference to the start time of $t_{\text{end}} - t_{\text{start}} = 2$.

Speedup Methods	t/timestep in s	t/Edge in ms	t/Node in s	Speedup
No Speedup	235	16,9	0,60	1,0
Left-hand difference	131	9,4	0,34	1,8
Zero Component Search	118	8,5	0,30	2,0
Inexact Newton	69,5	5,0	0,18	3,4
All together	40,5	2,9	0,10	5,8

Tab. 3.4: Speedup Methods for constructing the Jacobi matrix. Average time per iteration step t is listed. For the average, ten timesteps were calculated. Also, time per node (inner nodes at actual hypersurface) ant time per edge (all edges in triangulation according to applied algorithm 5) is shown. Times are noted in units of seconds (s) or milliseconds (ms) respectively. The speedup is calculated by dividing the time of the optimized method by the time of the method with no speedup. Note: When estamating the speedup factor with the values in the tables by hand, deviations occur due to roundoff errors.

Tab.(3.4) shows the speedup applying all three different approaches seperately and applying all methods simultaneously. In general when normalizing the time to compare different grids, it makes sense to distinguish between nodes and edges in the case of unstructured grids. This is because there is no fixed relation between inner nodes and number of treated edges anymore as it is the case with structured grid approaches. In all tests considered the result of the calculation deviates only within linewidth when plotted into a diagram. For special problems it may occur that the Inexact Newton method is not as stable as the ordinary Newton method. Therefore, when a breakdown occurs we are supposed to double check the result with an ordinary Newton method. Contrary to the Inexact Newton method and finite differences of lower order respectively, the Zero Component Search is exact and can be activated in every case.

Summarizing, we see a significant speedup factor of around 6 when all methods are applied together. The calculation was run at a 2.0 GHz XEON CPU with 6 megabyte of L2 cache.

3.4 Investigated Problems and Numerical Results

In this section, the for this thesis developed code relying on the algorithms presented in the preceeding section was applied to several problems. The section starts with a two-dimensional testbed. The curvature of a 2-sphere is calculated by means of the Time-Evolution scheme but by applying different equations than the Regge equations. With this low-dimensional testbed the evolution of the error of the equations is measured as well as convergence to the analytical solution when refining the grid.

From the second part on, the Regge equations will be considered only. It starts with a theoretical view on the problem of a spatial boundary. To my best knowledge, no discussion of the boundary of a domain together with the Time-Evolution scheme was undertaken in scientific publications till now. In closed universes, this problem does not appear since the domain is a 3-torus. But a lot of problems are modelled on a domain which has boundaries. The found relations were applied successfully to a cube domain in the Schwarzschild universe.

In the third part, two different numerical methods are compared in a framework of almost flat spacetimes. This means, the solution to the problem is dominantly a flat background metric plus a deviation from it which is of magnitudes smaller. Furthermore, the squared amplitude of the deviation vanishes within numerical accuracy. Two testbeds of the Apples-With-Apples testsuite were applied. First, the Robust-Stability test and a modified version from the Robust-Stability test on an unstructured grid. Second, the Linear-Wave test was considered. In all tests, the QR method was compared to the LU method. The LU method suppresses four Regge equations per point due to an assumed simplicial Bianchi identity. The QR method does not suppress any equations. In the present papers on Regge Calculus, to my best knowledge, QR method and LU method were never treated separately and the LU scheme was used mainly.

Afterwards, convergence of Regge Calculus is shown by the Linear-Wave test. New error measures were introduced coming from Finite-Element theory. They were applied to the 3-metric on each hypersurface which was reconstructed from the squared edge lengths. This approach is new to the field of Regge Calculus. Furthermore, a Courant condition was identified which coincides with the demand for causality.

The fourth part has a view on dominantly non-linear spacetimes in Regge Calculus. It was investigated how good the analytical solution is regained with the Time-Evolution scheme. Furthermore, the ability of lapse and shift to control the Time-Evolution was shortly investigated. The two spacetimes calculated are the Kasner universe and the Gowdy universe. By means of the Kasner universe it is again shown that a Courant condition holds true which corresponds to causality directly.

The fifth part closes with the calculation of the stationary Schwarzschild problem on an excised domain without applying any symmetries. The triangulation of the domain was generated with the external mesh generator `NETGEN`. For the first time such an approach was considered and calculated.

Before we start, an additional prerequisite is necessary to describe the calculation in detail. The Time-Evolution scheme can only start when the initial data is known and the corresponding squared edge lengths have been calculated. The initial data comprises data on two consecutive hypersurfaces and the edges between these both. To calculate the squared edge length exactly we must integrate the infinitesimal line element along the geodesic path connecting both end points of the edge. In this work two simplifications were made which simplify the problem. First, a sufficiently fine triangulation was assumed, so that the geodesic can be resonably well approximated with a line. Second, the line element was integrated by the mid point rule which multiplies the squared coordinate length of the edge with the metric corresponding to the midpoint of the edge. To give justification to this approach the result of the midpoint rule was compared to the result of the trapezodial summation rule with around 50 grid points. In all cases the initial values deviated only slightly. For a diagonal metric the mid point rule as an approximation to the curve integral along a geodesic can be calculated with:

$$
\begin{aligned}
l_{AB} &= \int_0^1 \sqrt{g_{\mu\nu}(\mathbf{x}(\lambda))\frac{\mathrm{d}x^\mu}{\mathrm{d}\lambda}\frac{\mathrm{d}x^\nu}{\mathrm{d}\lambda}}\mathrm{d}\lambda \\
&= \int_0^1 \sqrt{g_{\mu\mu}(\mathbf{x}(\lambda))(x_B^\mu - x_A^\mu)^2}\mathrm{d}\lambda \\
&\approx \sqrt{g_{\mu\mu}(\frac{\mathbf{x}_B - \mathbf{x}_A}{2}) \cdot (x_B^\mu - x_A^\mu)^2} \\
\Rightarrow \quad l_{AB}^2 &= g_{\mu\mu}(\frac{\mathbf{x}_B - \mathbf{x}_A}{2}) \cdot (x_B^\mu - x_A^\mu)^2 \,.
\end{aligned}
$$

Here, A and B denote the vertices of the edge and $x^\mu(\lambda)$ parametrizes the geodesic. The derivation of the geodesic was replaced with $(x_B - x_A)$ since the line is parametrized with $x^\mu(\lambda) = \lambda x_B + (1-\lambda)x_A$.

3.4.1 Convergence and Evolution Error with Alternative Equations for Two Dimensions

For code testing lower dimensional equations are of interest. The Regge equations are non-trivial in four dimensions only. To get started in code writing and to get a feeling about the Time-Evolution scheme it is a good idea to have a look at some related simplicial methods which make sense in two dimensions. In two dimensions the bone is equivalent to a point P. Further, let us restrict the problem to the Euclidean signature. Then it is sufficient to treat the ordinary edge lengths as variables, rather than the squared edge lengths. The scalar curvature R_p at point P is defined as

$$
R_p := \frac{2}{3}\frac{\varepsilon_p}{A_p}\,, \quad A_p = \sum_{t|p\in t} A_t\,, \tag{3.30}
$$

where ε_p is the deficit angle at point P and A_p is the cumulated area of all triangles surrounding point P. Therefore, by knowing R_p for each point P one non-linear equation per point can be stated. Considering the grid illustrated in Fig.(3.18), there is only one edge length x unknown. Both other new edges are already set by lapse l_L and shift l_S. That means the number of unknowns and the number of equations are both equal to one.

To state analytic equations a serial evolution is reasonable. That means the points in the one-dimensional subspace are evolved one after another. Every time the lapse and shift edges stay the same relatively to the evolution point. So the problem for each point is always equivalent to Fig.(3.18). Besides a lot of other methods of how to solve this one equation, a Newton scheme as discussed in section 3.1.5 can be employed. There, the derivation $\partial R_p/\partial x$ is needed to set up the Newton method. The necessary terms for this derivation are

$$
\begin{aligned}
\frac{\partial \varepsilon_p}{\partial x} &= \frac{\partial}{\partial x}\left(2\pi - \sum_i \alpha_i\right) \\
&= -\frac{\partial \alpha_5}{\partial x} = -\frac{\partial}{\partial x}\arctan\frac{x}{l_L} = \frac{l_L}{{l_L}^2 + x^2}\,, \\
\frac{\partial A_p}{\partial x} &= \frac{\partial}{\partial x}\left(2\left(x^2 l_L^2 + x^2 l_E^2 + l_L^2 l_E^2\right) - x^4 - l_E^4 - l_L^4\right) \\
&= \frac{x}{8A_p}\left(l_L^2 + l_E^2 - x^2\right)\,.
\end{aligned}
$$

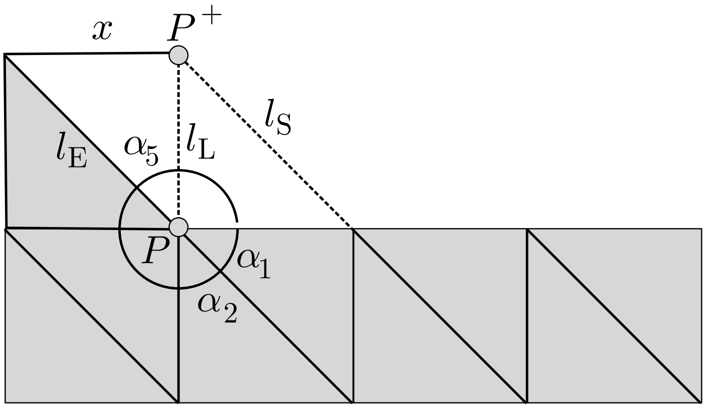


Fig. 3.18: 2d grid for solving 2d simplicial equations. One new equation arise for the evolved point P. Three new edges arise (l_S, l_L, x). Two of them must be fixed (dashed lines) to fix point P^+ in the geometry. One equation for one unknown (x) is left.

Summarizing, there are two main differences to the four-dimensional Regge Calculus when dealing with this approach. First, the resulting number of equations and number of unknowns are equal *after* fixing lapse and shift. Therefore a quadratic system has to be solved. Second, the equation Eq.(3.30) rely directly on the bones p, the geometrical object where the curvature resides. Here, each bone gives one equation. Contrary to the four-dimensional Regge Calculus. There, each edge gives one equation and the bones are the triangles.

The 2-Sphere

A 2-sphere can be described with two coordinates $\phi \in [0, 2\pi]$ and $\theta \in [0, \pi]$. The corresponding line element is,

$$\mathrm{d}s = r^2\,\mathrm{d}\theta^2 + r^2\,\sin^2\theta\,\mathrm{d}\phi^2\,.$$

Here, r is the radius of the 2-sphere when embedded in a flat three-dimensional space. A 2-sphere has constant curvature. After introducing a triangulation $\mathcal{T}$ of the 2-sphere the scalar curvature R_p is located in each point p and is constant for every point:

$$R_p = \text{const.} = \frac{2}{r^2}\,, \qquad \text{for all}\, p \in \mathcal{T}\,.$$

With knowing R_p the method from section 3.4.1 can be applied to evolve a given initial triangulation in one coordinate direction. Here, the ϕ-direction was chosen to be the evolution direction.
Before evolution can start, initial data is posed on the domain $[\frac{\pi}{4}, \frac{3\pi}{4}] \times [0, \Delta\phi]$ with a structured two-dimensional grid shown in Fig.(3.18). The stepsize in both directions was chosen to be equal: $\Delta\phi = \Delta\theta$. The evolution scheme does not make use of point classes. Instead, a serial evolution was employed as shown in 3.4.1. Boundary edges were set with respect to the analytical edge length. The problem is solved with an one-dimensional Newton-Raphson method. It is very stable, even if the start value is perturbed strongly.

To investigate the convergence rate with respect to the step size the geometry was evolved in ϕ-direction one circulation from $\phi = 0$ to $\phi = 2\pi$ with radius $r = 100$. Anticipating the error

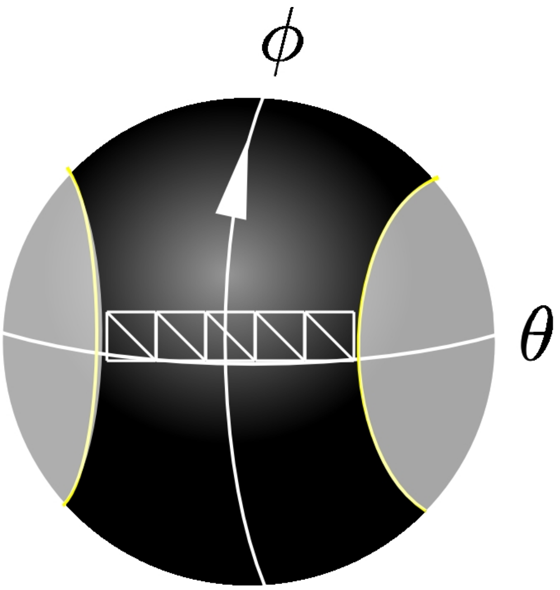


Fig. 3.19: A 2-sphere. The yellow lines highlight the boundaries. The initial data is sketched with white color. The evolution direction is shown with a white arrow.

measures for the four-dimensional Regge Calculus, analogue error measures were constructed at this place to make a comparison possible. The aim is to replace the 3-metric with the appropriate 1-metric. In the example of the 2-sphere, the metric tensor corresponding to the hypersurface reduces to one component $g_{\theta\theta}$. In this case the relation between metric and squared edge length becomes:

$$l_{\theta\theta}{}^2 = g_{\theta\theta} \cdot (\Delta\theta)^2 \,. \tag{3.31}$$

Now we are able to construct error measures based on this metric component. A detailed list of already used error measures for Regge Calculus in previous work is listed in the next section. Instead, in this work, we choose error measures which are well known in Finite-Element theory and have a strong analytical background. This is the infinity norm and the L_2 norm. By defining the pointwise error as

$$u(\theta,\phi) := g_{\theta\theta}^{\mathrm{num}}(\theta,\phi) - g_{\theta\theta}^{\mathrm{an}}(\theta,\phi) \,. \tag{3.32}$$

these error measures are supposed to calculate the deviation of the numerical solution $g_{\theta\theta}^{\mathrm{num}}$ to the analytical expected value $g_{\theta\theta}^{\mathrm{an}}$. The following definitions hold:

$$\begin{aligned} ||u(\cdot,\phi)||_{\infty,\Omega_\phi} &:= \max_\theta |u(\theta,\phi)| \\ ||u(\cdot,\phi)||_{L_2,\Omega_\phi} &:= \sqrt{\int_{\Omega_\phi} |u(\theta,\phi)|^2 \mathrm{d}\theta} \,. \end{aligned} \tag{3.33}$$

Here, the symbol Ω_ϕ emphasizes that not a two-dimensional domain is considered but instead the one-dimensional hypersurface at fixed ϕ value is taken. In the present example $\Omega_\phi = [\pi/2, 3\pi/2]$ holds. To keep things clear the domain Ω_ϕ is suppressed in the notation of the error norms in plots later on.

When we consider the corresponding relative error to Eq.(3.32) and use Eq.(3.31) we get the following relation which can be used to generate formulas to calculate Eq.(3.33):

$$\begin{aligned}\frac{u(\theta,\phi)}{g^{\mathrm{an}}_{\theta\theta}(\theta,\phi)} &= \frac{l^{\mathrm{num}2}_{\theta\theta} - l^{\mathrm{an}2}_{\theta\theta}}{(\Delta\theta)^2} \cdot \frac{(\Delta\theta)^2}{l^{\mathrm{an}2}_{\theta\theta}} \\ &= \frac{l^{\mathrm{num}2}_{\theta\theta} - l^{\mathrm{an}2}_{\theta\theta}}{l^{\mathrm{an}2}_{\theta\theta}}\end{aligned}$$

We see, that we need only the analytical squared edge length $l^{\mathrm{an}2}_{\theta\theta}$ and the numerical squared edge length $l^{\mathrm{num}2}_{\theta\theta}$ to calculate the relative error of the metric component $g_{\theta\theta}$. This is true if we have only one metric component. In the four dimensional Regge Calculus analogue equations can be stated only in special cases. The general approach to calculate these norms in four dimensions is presented in the next section.

In Fig.(3.20) both error measures are plotted versus the evolution coordinate ϕ. Both errors are normalized to the value of the norm of the analytical solution $||g^{\mathrm{an}}_{\theta\theta}(\cdot,\phi)||$ and $||g^{\mathrm{num}}_{\theta\theta}(\cdot,\phi)||$ respectively. We see a linear raise in both cases. Several errors corresponding to different discretization step widths $\Delta\phi = \Delta\theta = \pi/(2N)$ are plotted. The third plot in the figure shows both error measures after one circulation: $\phi = 2\pi$. Also, in this logarithmic plot, a polynom of order two is plotted showing that both error measures decrease with quadratic order. This can be summarized with:

$$\begin{aligned}||u(\cdot,\phi)||_{\infty,\Omega_\phi} &:= \mathcal{O}((\Delta\phi)^2) + \mathcal{O}((\Delta\theta)^2) \\ ||u(\cdot,\phi)||_{L_2,\Omega_\phi} &:= \mathcal{O}((\Delta\phi)^2) + \mathcal{O}((\Delta\theta)^2)\,.\end{aligned}$$

In the next test is was examined how long the code runs maximally and under which circumstances the code crashes. It can be expected, that the code crashes when the error is of the same magnitude as the squared edge length itself. To see this, an alternative measure was employed, namely the sum of the deviation of the numerical squared edge length to its analytical counterpart. With $N = \frac{\pi}{2}/\Delta\theta$ a coarse grid with $N = 10$ and a fine grid with $N = 40$ were considered. For $N = 10$, the magnitude of the squared edge lengths is $l_i{}^2 \approx 1\ldots 5\cdot 10^2$. For $N = 40$, $l_i{}^2 \approx 10^1$. The result is plotted in Fig.(3.21). We see, when the error is close to the magnitude of the squared edge length the code crashes. With the fine grid the code runs four times longer till $\theta \approx 80$ while the coarse case crashes around $\theta \approx 20$.

Discussion of Results

Summarizing the results of the introductory test of the Time-Evolution scheme applied to alternative equations in two dimensions, the Infinity norm and the L_2 norm of the error defined in Eq.(3.32) and Eq.(3.33) respectively follow the estimation

$$\mathrm{Error} = \mathcal{O}((\Delta\theta)^2) + \mathcal{O}((\Delta\phi)^2)\,.$$

Furthermore, the error growths linearly with increasing time.

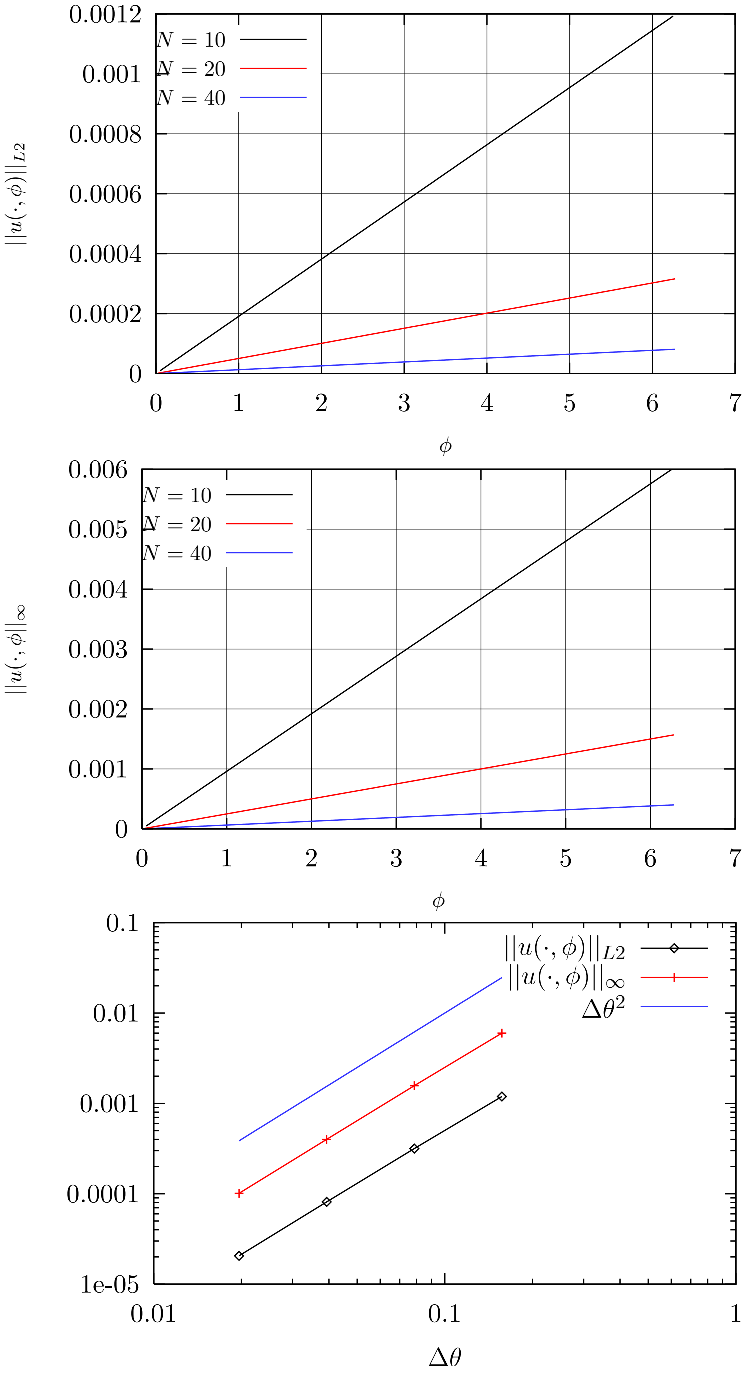


Fig. 3.20: 2-Sphere. Convergence of infinity norm and L_2 norm of the metric component $g_{\theta\theta}$. *Top and Middle.* Both error measures plotted versus evolution coordinate ϕ. A linear growth in ϕ can be observed. *Bottom.* Error after one circulation ($\phi = 2\pi$) is plotted versus different discretization step width $\Delta\theta$. In this logarithmic plot a quadratic convergence can be detected.

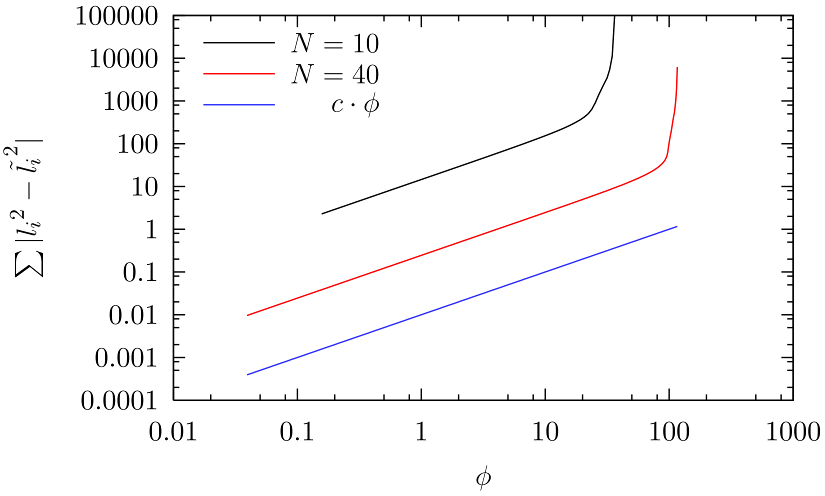


Fig. 3.21: Not-normized error versus evolution coordinate ϕ. For two different grid resolutions a linear raise of the error occurs. This can be verified with the blue line in the logarithmic plot.

3.4.2 Spatial Boundaries and Methods to fix Lapse and Shift

Spatial Boundaries in Regge Calculus

In the theoretical framework of Regge Calculus the problems which occur near to the boundary are usually skipped. This section deals with phenomenas which occur at the boundary of the triangulation. First, let us assume for the sake of simplicity that $\partial\Omega = \partial\mathcal{T}$, that means the boundary of the domain falls together with the boundary of the triangulation. When looking at domains with a boundary $\partial\Omega$ not every single edge e_i gives rise to an equation G_i. When a bone is part of the boundary ($b \in \partial\Omega$) it does not have a meaningful deficit angle. In this case, the bone is not completely surrounded with simplices, that means a vector can not be parallel transported around all closed loops in the neighborhood of this point. This is maybe most obvious in the image space $\mathcal{M}$. There, outside the domain, triangles are missing to calculate a deficit angle. Which effects does this have considering the Regge equations? Obviously, since the Regge equation is composed of terms corresponding to bones hinging at the edge, all bone terms must be meaningful. If a Regge equation corresponding to an edge is invalid, a bone can be constructed with two additional edges so that this bone is part of the boundary. Since $e_i \subset b$ this happens only if the edge itself is part of the boundary, $e_i \in \partial\Omega$. In this way, for spacetime domains with a spatial boundary we can state:

I Each edge $e_i \notin \partial\Omega$ is associated to a Regge equation G_i.

II Each boundary edge $e_i \in \partial\Omega$ gives no equation since there exists at least one bone for which $b \in \partial\Omega$. Therefore, the necessary quantity ε_b can not be determined for this bone.

Usually, the squared edge lengths of boundary edges are determined by boundary conditions. This gives us a kind of Dirichlet boundary condition for the spatial boundary of spacetime.

But while applying the Time-Evolution scheme with points inside the domain only, not every edge inside the domain is treated. By applying the Time-Evolution scheme to boundary points $A \in \partial\Omega$ edges inside the domain corresponding to this point are calculated. These edges for which $\overline{AP}$ with $P \notin \partial\Omega$ holds true correspond to a new equation or new unknown. Fig.(3.22) illustrates the problem with a two-dimensional sketch. The picture holds true in four dimensions if the prism method in timelike direction is applied. We can see, that characteristical equation-unknown pairs arise. In the figure, the inner point P has already been evolved. The Time-Evolution scheme for point A demands for the squared edge lengths $l^2(\overline{AA^+})$ and $l^2(\overline{A^+P^+})$. The first one is prescribed because it is part of the boundary (Dirichlet boundary condition). For the remaining edges inside the domain corresponding to the Time-Evolution of a boundary point two different methods can be applied:

(A) Solve equation corresponding to $\overline{AP^+}$ to get squared edge length $l^2(\overline{A^+P^+})$,

(B) Prescribe $l^2(\overline{A^+P^+})$ with an additional boundary condition, so that the equation to $\overline{AP^+}$ is fulfilled automatically.

In numerical experiments undertaken, method B was more stable in almost all cases. There, no additional non-linear system of equations must be solved, so this result can be expected to a certain amount.

Summarizing, when the spacetime has a spatial domain, then first the edges on the boundary must be initialized with a Dirichlet boundary condition. Second, the Time-Evolution scheme for boundary points give rise to new equations and unknowns which lie inside the domain.

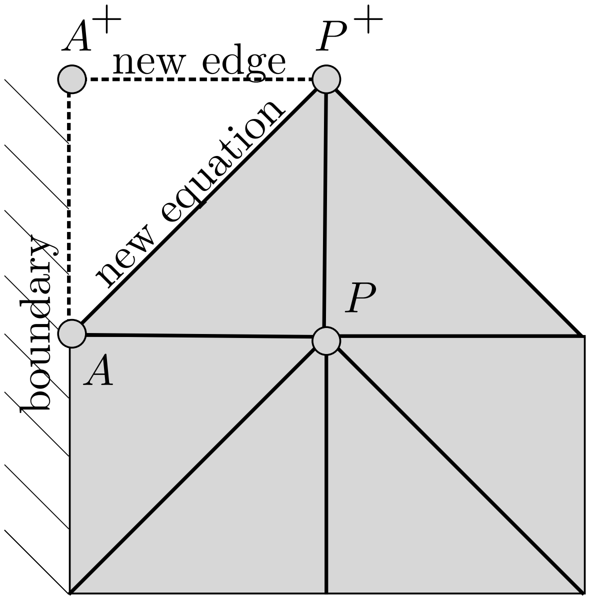


Fig. 3.22: When evolving boundary point A new equations arise. This 2d sketch shows one unknown edge and one edge corresponding to an equation. *Either* the new edge is calculated by the new equation *or* the new edge is prescribed using boundary conditions.

This can be best described with equation-unknown pairs. The unknown edge, the edge corresponding to an equation and the lapse edge give us a triangle geometrically. The unknowns and equations corresponding to a boundary point give us a system of equations which can be solved numerically or by posing an appropriate additional boundary condition.

In this work method B is applied and the explicit method chosen goes back to [Gal93]. Given a triangle with a timelike edge and given two edge lengths of this triangle, then the third egde length can be calculated assuming flat metric. This is justified by the assumption that the triangulation is sufficiently fine.

So we take each equation-unknown pair of a boundary point together with the lapse edge. This gives us the mentioned triangle. Now, we calculate the unknown edge length by means of the *Pythagorean Theorem*. Let $a^2 < 0$ denote the squared edge length of the lapse, $b^2 > 0$ denote the squared edge length of the edge corresponding to the new equation and c^2 denote the unknown squared edge length, then:

$$c^2 = \begin{cases} b^2 + a^2 & \text{if } c^2 < b^2\,, \\ b^2 - a^2 & \text{if } c^2 > b^2\,. \end{cases} \tag{3.34}$$

Fig.(3.23) shortly denotes this idea and it becomes clear that $c^2 < b^2$ or $c^2 > b^2$ respectively is known a priori. This method will be referred to as *Galassi method.* This approach works exactly in the case of a flat metric and it is a good approximation in non-flat cases if the triangulation is sufficiently fine.

Two Methods to Fix Lapse and Shift

If we want to fix lapse and shift edges two different approaches were applied in this work. First, we can apply the Galassi method. For example if the lapse edge is supposed to emanate

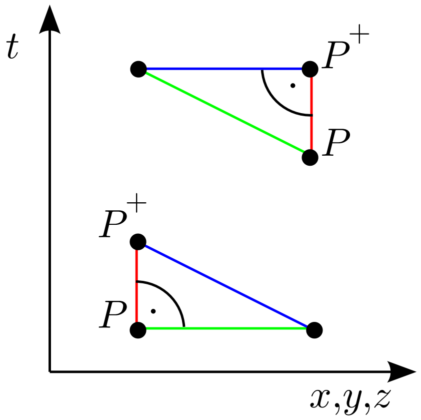


Fig. 3.23: Galassi method to fix unknown edges. With the lapse edge (red) and the equation edge (green) the unknown edge (blue) is easily determined, assuming flat metric.

perpendicular to the hypersurface Eq.(3.34) can be used. Then the Galassi method starts by fixing first the lapse edge. Then three shift edges which should all belong to the same 3-simplex for stability reasons are fixed with the Pythagorean Theorem. In this way the lapse edge emanates perpendicular only to this 3-simplex. For the special case of a flat metric the lapse edge also emanates perpendicular to all other 3-simplices surrounding the lapse edge but in general, the lapse edge will emanate perpendicular to the chosen 3-simplex only. This special simplex can be chosen at will for the inner points in principle. But numerical tests did show that for inner points which are connected to the boundary with an edge, a special 3-simplex is chosen for the reason of stable evolutions. The 3-simplex must hinge at the timelike edge and the 3-simplex must have the most edges connecting P with the boundary.

Summarizing, we can state the following recipe:

(1) Set edges $e \in \partial\Omega$ according to a boundary condition.

(2) Set equation-unknown pairs for Time-Evolution points $P \in \partial\Omega$ with the Galassi method.

(3) If $P \notin \partial\Omega$ and new unknown edges *do not* touch boundary, then apply Galassi method to set shift edges. Choose one arbitrary 3-simplex.

(4) If $P \notin \partial\Omega$ and new unknown edges *do* touch boundary, then apply Galassi method to set shift edges but choose the 3-simplex with the most edges connected to the boundary for that.

The second method to fix lapse and shift is called *Zero-Shift conditions.* Again, the lapse edge is chosen with respect to each special problem. Then we apply structured grids with a point order that give reason to triangles shown in Fig.(3.24). Assuming PP^+ as the timelike edge in the Time-Evolution scheme opposite points of P^+ give rise to opposite edges which are demanded to be equal in all three spatial directions.

According to the illustration of a structured triangulation in Fig.(3.24) we can start from point P in $x-$, $y-$ and $z-$direction and find neighbor points P_x^-, P_x^+, P_y^-, P_y^+, P_z^- and P_z^+.

Then the Zero-Shift conditions demand:

$$\begin{aligned}\overline{PP_x^-} &= \overline{PP_x^+}\\ \overline{PP_y^-} &= \overline{PP_y^+}\\ \overline{PP_z^-} &= \overline{PP_z^+}\,.\end{aligned} \tag{3.35}$$

Either the shift edges must be set to the appropriate value after each iterations step or the equations Eq.(3.35) are solved together with the corresponding system of Regge equations. In this work, the latter method was applied. We see that the lapse edge $\overline{PP^+}$ emanates perpendicular to the hypersurface corresponding to edges which do not belong to only one 3-simplex.

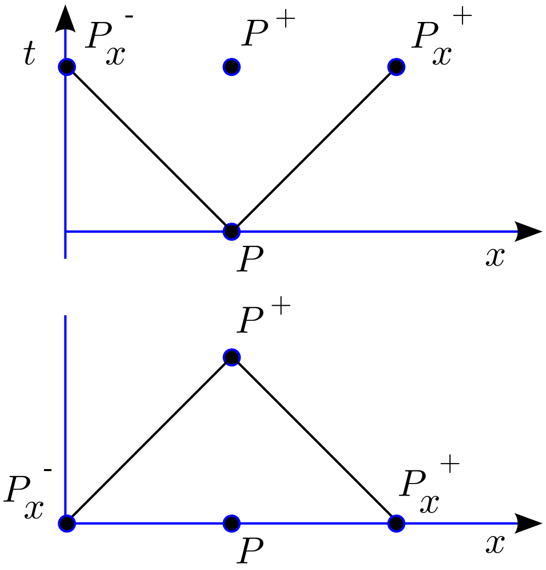


Fig. 3.24: Zero-Shift conditions. Opposite edges are demanded to have the same edge length.

For grids with an arbitrary point order the Zero-Shift conditions are not applicable directly since additional triangulations than shown in Fig.(3.24) can occur. It may occur that P_x^+ is in hypersurface $t = t_0$ and P_x^- is in hypersurface $t = t_1$. Then a constraint equation like $\overline{PP_x^+} = \overline{PP_x^-}$ is useless. The appearance of such triangulations is a general property of the prism method. It can be circumvented with structured grids and an appropriate point order to evolve the three-dimensional grid. With unstructured grids a feasible graph coloring is a tool to make the Zero-Shift conditions applicable for almost all points.

Test: Schwarzschild Spacetime

In this test, the metric given by a static black hole in vacuum is investigated. As introduced in section 2.4.1 the metric in isotropic coordinates and $M = 1$ was chosen:

$$\mathrm{d}s^2 = -\left(\frac{1-1/2r}{1+1/2r}\right)\mathrm{d}t^2 + \left(1+\frac{1}{2r}\right)^4 (\mathrm{d}r^2 + r^2\,\mathrm{d}\theta^2 + r^2\sin^2\theta\,\mathrm{d}\phi^2)\,. \tag{3.36}$$

The black hole singulartiy occupies the origin of the coordinate system. The metric is spherical symmetric and static, meaning that the metric depends on the radius coordinate only. The sphere at $r = 1/2$ is called event horizon. The proper time for an observer on the event horizon vanishes as can be seen by considering the metric element g_{tt}. The spatial metric is conformal flat. That means, the metric is a flat metric multiplied with a conformal factor. Here, this factor is equal to $\psi^4 = (1 + 1/2r)^4$. At the event horizon $\psi = 2$ and in the limit to infinity $\psi = 1$.

In this test the Schwarzschild spacetime is calculated on a supercube $[2, 2.3]^3$ in Cartesian coordinates. The supercube consists of $3 \times 3 \times 3$ single cubes which were triangulated with the modified scheme of Gentle and Miller according to section 3.2.2. In short, we have to project points of the bumps onto the six faces of the supercube which occur when not closing the triangulation of Gentle and Miller. Edges which are part of the six faces of the supercube were initialized with respect to the given analytical metric Eq.(3.36). It was outlined that the Time-Evolution scheme applied to boundary points give rise to additional edges inside the domain which must be initialized appropriately for each time step. Let $\mathcal{E}$ denote this set of all edges which start at the boundary and then reach into the domain:

$$e \in \mathcal{E} \quad :\Leftrightarrow \quad e = \overline{AB} \quad \text{with } (A \in \partial\Omega \text{ and } B \notin \partial\Omega) \text{ or } (B \in \partial\Omega \text{ and } A \notin \partial\Omega)\,. \tag{3.37}$$

Here, we prescribe all edges from $\mathcal{E}$ by means of an appropriate equation. Two approaches will be tested which should point out the importance of handling these edges:

First, edges from $\mathcal{E}$ are not treated in any way, instead they are kept constant. At first glance, this could succeed since the Schwarzschild metric is static. Second, edges from $\mathcal{E}$ are initialized with the Galassi method. In both cases, the edges $e \in \partial\Omega$ are set to its analytical value. The timestep is fixed and was chosen to be $dt = 0,01$.

Here, two different error measures (EM) were considered. The first error measure (EM1) is the averaged absolute deviation from numerical squared edge length to analytical squared edge length. The second error measure (EM2) sums up the absolute value of each Regge equation. In this way it represents the residual of the system of equations. The result of this test is shown in Fig.(3.25).

The main essence of this figure is that edges from $\mathcal{E}$ which are neither treated by Time-Evolution schemes for inner points nor treated with a Dirichlet boundary condition must be initialized appropriately. It is not possible to neglect these edges and get long-time evolutions. By applying the *Galassi method* to these edges the code runs till $t \approx 40 = 4000$ timesteps. By neglecting these edges the code crashes within a few timesteps. The absolute discrepancy to the exact solution, raises linearly till $t \approx 20$ and then increases of higher order till the code crashes. Also, the sum of Regge equations raises linearly till $t \approx 20$ and then show strong oscillating behavior which indicates that a crash is coming up.

With this test we did see that boundary conditions have a tremendous impact on Time-Evolution with Regge Calculus. Appropriate conditions can be stated to get long-term Time-Evolutions.

3.4.3 The QR Scheme to Solve Almost Flat Spacetimes

In this section almost flat spacetime are investigated with the subject to highlight the difference between solving the Regge equations with the QR method and by solving them with the LU method taking constraint equations into account. What means this in detail? First,

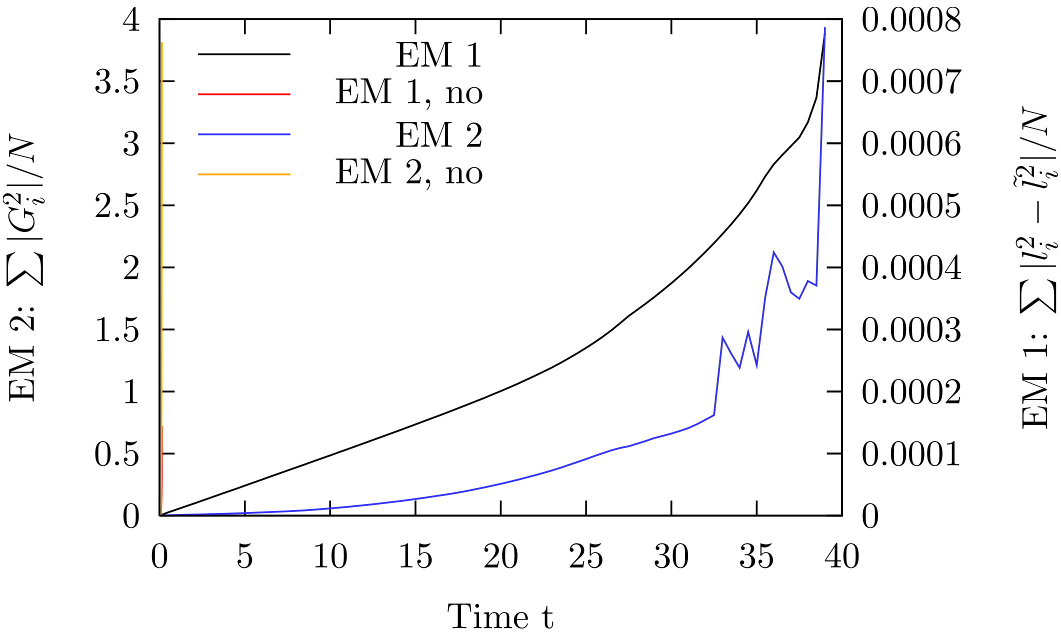


Fig. 3.25: Schwarzschild Evolution. EM = Error Measure. no = Squared edge length of edge/equation-pair on boundary are set to constant value, Else: Galassi method is applied to these edges.

with *almost flat spacetime* metrics are treated which are dominantly flat Minkowski metric. Only to a small extent and ways of magnitude smaller non-linear terms occur in the mertric. The non-linearity is chosen with an amplitude which vanishes in numerical accuracy when taking the square of the non-linear term. Second, the difference between the application of the QR method and the LU method is the consideration of a simplicial Bianchi identity. As developed in the preceeding sections the system of non-linear equations corresponding to the Time-Evolution of a point is rectangular meaning that the number of unknowns is lower than the number of equations. Usually, as in all considered triangulation schemes, the number of unknowns is equal to the number of equations minus four: $N_{\text{unknowns}} = N_{\text{edges}} - 4$. This system is overdetermined and non-linear. The Newton-Raphson method transfers the problem of solving a non-linear system to solve linear system of equations successively. Linear, overdetermined systems of equations are solved with the QR method in this work. A LU method is applied whenever the number of equations match the number of unknowns.

The LU Method

This method is standard and explained in a broad range of books on numerics, for instance [DH03]. It is often mentioned together with the *Gaussian Elimination*. Given the problem $Ax = b$, the matrix A is decomposed in the matrix product LU where L is a lower triangular matrix and U is an upper triangular matrix. Then the problem is solved by first solving $Lz = b$ and second solving $Ux = z$. This results in forward substitution and backward substitution of equations and can be solved easily with an algorithm. The main effort N^3 is to decompose the matrix. Here, N is the number of equations or unknowns respectively. Despite this high effort it is unproblematic in the Time-Evolution scheme since the number of equations or unknowns are very low compared to other problems in scientific computing. Usually, around 15 equations occur depending on the used triangulation. The LU method can be applied to quadratic systems only.

For the LU method the following termination criterion is chosen. With the target function F defined as

$$F(x) = \frac{1}{2}\langle \mathbf{G}(x), \mathbf{G}(x) \rangle \, ,$$

the termination criterion becomes

$$F(x^k) < \mathrm{TOL} \qquad \text{Termination Criterion, LU method} \; . \tag{3.38}$$

The LU method is exact in the sense that its solution is exact within numerical accuracy. In the different numerical experiments undertaken in this work TOL was chosen with appropriate magnitude. Often, TOL $\approx 10^{-24}$ managed to work well.

The QR method

Given the problem $Ax = b$, the $m \times n$-matrix A is decomposed in the matrix product QR, where Q is an orthogonal $m \times m$-matrix ($Q^T Q = 1$) and R is an upper triangle $m \times n$-matrix. Having done this decomposition, the approximate solution to the rectangular system is obtained by $x = R^{-1}\tilde{b}$, where $\tilde{b}$ is the vector consisting of the first n elements of the m-dimensional vector $Q^T b$. Then the solution is approximate in the sense that it minimizes the residual $||b - Ax||$ with respect to the Euclidean norm. This means the QR method solves a *linear least-squares problem.* The Newton method applied is an unconstrained Gauss-Newton algorithm. There, a target function is minimized with respect to the Euclidean norm. The target function F is chosen as:

$$F(x) = \frac{1}{2}\langle \mathbf{G}(x), \mathbf{G}(x) \rangle \, .$$

The vector $\mathbf{G} = (G_0, G_1, \ldots, G_{N_{\text{equations}}})$ collects all Regge equations contained in the local Time-Evolution scheme. The symbol x collects all dependencies on squared edge lengths of the triangulation. These are all edges of all simplices hinging at the timelike edge. Then, the Gauss-Newton scheme converges to a solution x^* for which

$$||F(x^*)||_2 = \min_x ||F(x)|| \, ,$$

if the method can be applied successfully. The termination criterion used in this case is:

$$|F(x^{k+1}) - F(x^k)| \leq \mathsf{TOL} \qquad \text{Termination Criterion, QR method} \, , .$$

It is also common to track the convergence of the Gauss-Newton method with a convergence monitor to determine divergence. Whether this can easily be applied in any code, it was not useful in the numerical simulations in this thesis. The signature of each simplex stated a much higher restriction to the method than a convergence monitor during the simulation. In the case of a non-convergent Newton scheme the numerical simulation usually breaks down immediately due to an observed signature failure of one simplex in the triangulation.

The popular book about numerical algorithms [TVF02] recommends a singular-value decomposition to solve an overdetermined and linear system of equations. But here I found the QR method simpler to derive. Also it is part of a broad range of books on numerics, for instance I can recommend [DH03]. Both methods give identical solutions within numerical accuracy.

Bianchi Identities

Every work on the Time-Evolution scheme considered for this thesis deals with a quadratic system of equations instead of the formerly mentioned overdetermined system of equations. At least no work investigates the difference between LU method and QR method. Why is that? First, R.SORKIN treated $N_{\text{eq}} \times N_{\text{edges}}$ systems only (see [Sor74]) where N_{eq} is the number of new equations and N_{edges} is the total number of new edges. In this way the new point in the geometry is not fixed and the problem is not well posed as discussed in Fig.(3.4). This is a quadratic system but it lacks lapse and shift conditions. Later M. GALASSI showed in [Gal93] the need for lapse and shift which is equivalent to the idea of fixing the point in the geometry as discussed in this work. Now, the resulting system would be rectangular but it is further argued that four equations of the N_{eq} new equations are *constraint equations.* If the other $N_{\text{eq}} - 4$ equations are solved, then these constraint equations should be satisfied automatically. The reason comes from the contracted Bianchi identity in continuum,

$$\nabla^{\alpha} \cdot G_{\alpha\beta} = 0\,.$$

The Einstein tensor $G_{\alpha\beta} = R_{\alpha\beta} - 1/2 R g_{\alpha\beta}$ is divergence free in all four indices. Therefore, four relationship between components of $G_{\alpha\beta}$ can be stated. These equations are usually called constraints. In [Mil86] it is argued that there are also four constraint equations per vertex in Regge Calculus. Regarding the picture of equation-unknown pairs, the edge corresponding to a constraint equation is chosen to lie directly under the shift edge. For instance, if AP^{+} is the shift edge then AP is chosen to be constrainted. Also the equation corresponding to the timelike edge is constrainted giving four constrainted equations in total. As Galassi did show in [Gal93] the constraints can only be fulfilled approximately. Whether the deviation to the exact solution of the constrainted edges vanishes with finer grid resolutions, non-vanishing constraints occur in every simulation. This means that the LU method solves the problem only approximately. Whether the resulting quadratic system can be solved within numerical accuracy with the LU method, the Regge equations corresponding to constrainted edges have a much higher error in magnitude. At this point it can be argued that the constraint equations can be left away in favor of the QR method. Both approaches solve the problem approximately only. Also, there is no exact derivation for a simplicial Bianchi Identity applied to the Time-Evolution scheme. In [HK04] exact Bianchi identities for Regge Calculus are stated for the first time but no application to the Time-Evolution scheme was made.

Due to this lack of knowledge, in this work the full rectangular system was solved as well as a quadratic system with constrainted equations. Different tests were applied, namely the Robust-Stability test and the Linear-Wave test. Both tests are part of the Apples-With-Apples test suite. The rectangular system was solved with the QR method, the quadratic system was treated with the LU method.

The Robust-Stability Test

This section will deal with the so called *Robust-Stability test*, described in detail in [AAH^{+}03] and [BHW^{+}07]. This is the easiest non-trivial test case. Assuming Cartesian coordinates in space (x, y, z), it perturbs the spacelike direction by a small but significant amount:

$$\mathrm{d}s^2 = -\mathrm{d}t^2 + (1+\varepsilon_x)\,\mathrm{d}x^2 + (1+\varepsilon_y)\mathrm{d}y^2 + (1+\varepsilon_z)\,\mathrm{d}z^2\,. \tag{3.39}$$

Here ε_x, ε_y and ε_z are independent random numbers with the property that ε^2 is below numerical accuracy so that the geometry is mainly linear.

Test 1: The Robust-Stability Test in a Closed Universe

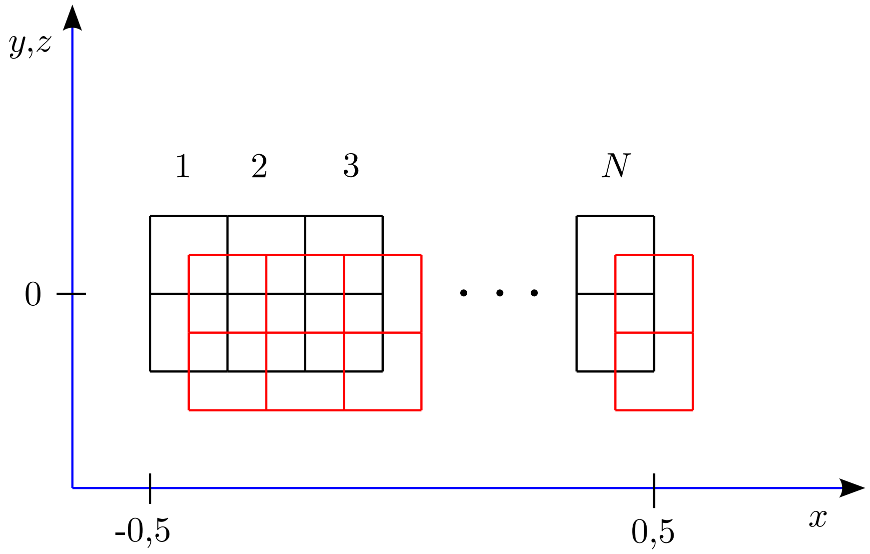


Fig. 3.26: Simplified triangulation of the domain. Triangulation of cube suppressed.

To test the metric Eq.(3.39) on a closed universe, the triangulation by Gentle and Miller was employed as introduced in section 3.2.2. All parameters from [AAH$^+$03] were taken over, that is:

(1) A minimal space triangulation in y- and z-direction was chosen so that the scheme does not collapse. Especially here, that means a slice at $x = \text{const.}$ which slices at least one point of point class A, slices four points of point class A in total. The same is valid for B-, C- and D-points. In the middle of the triangulation the following holds true: $y = 0$ and $z = 0$.

(2) $x \in [-0.5, 0.5]$. The x-axis is divided into N intervals.

(3) $\Delta x = 1/N$. The timestep is $\Delta t = \Delta x/4$.

(4) The perturbation is set to $\varepsilon \in (-10^{-10}/(\frac{N}{50})^2, +10^{-10}/(\frac{N}{50})^2)$.

The triangulation is sketched in Fig.(3.26). Due to the torus character of the triangulation no boundary issues must be challenged. To make a decision if the LU method or the QR method works better, the residual of the Regge equations was monitored which is defined in this thesis with

$$\text{Residual} = \frac{1}{N_{\text{edges}}} \sum_{\text{edge } i \,|\, i \notin \partial\Omega} |G_i| \,. \tag{3.40}$$

Eq.(3.40) measures the deviation from the exact solution of the Regge equations and is normalized to the number of equations which is equal to the number of inner edges N_{edges}.

Each Regge equation G_i is supposed to be zero but in the QR scheme the system is overdetermined and the equations can solved in a least-squares sense only. In the LU scheme, four equations are left aside due to the simplicial Bianchi indentity. These equations do not fulfill the Regge equations exactly. This was already discovered in [Gal93]. There, deviations from $G_i = 0$ concerning these constrainted edges were discovered. Of course, all remaining equations in the LU scheme are now solved exactly, since the resulting system of equations is quadratic. So, in both schemes, an error in the equations is expected and is measured by summing up the absolute values of all Regge equations. This value is referred to as *residual*.

In the used triangulation we have 15 equation-unknown pairs for each point to evolve. We have to fix four edge lengths. One of them is the timelike edge connecting the point to evolve P with its pendant one timestep further P^+. The three remaining edges are initialized by means of the Zero-Shift conditions as explained in the previous section.
The lapse edge or the timelike edge respectively has a special property. Its absolute squared edge length is much smaller than all spacelike squared edge lengths. Therefore, the area of all bones hinging at this edge are extremly small. This yield very high values for the Regge equation corresponding to the timelike edge because the area of the bone influences the value reciprocally. With this qualitative property in mind, not only the LU method and the QR method was tested here, but also a modified QR scheme. There, only the timelike edge is constrainted. Tests were run considering zero constraint equations (QR method), one constraint equation (timelike edge, modified QR method) and four constraint equations (Simplicial Bianchi identity, LU method). It turned out, that the QR method and the modified QR method give similar results.

Comparing LU method and QR method, the best results were achieved with the QR scheme. Fig.(3.27) shows long term results for the residual concerning the QR scheme. The residual keeps stable with respect to time.

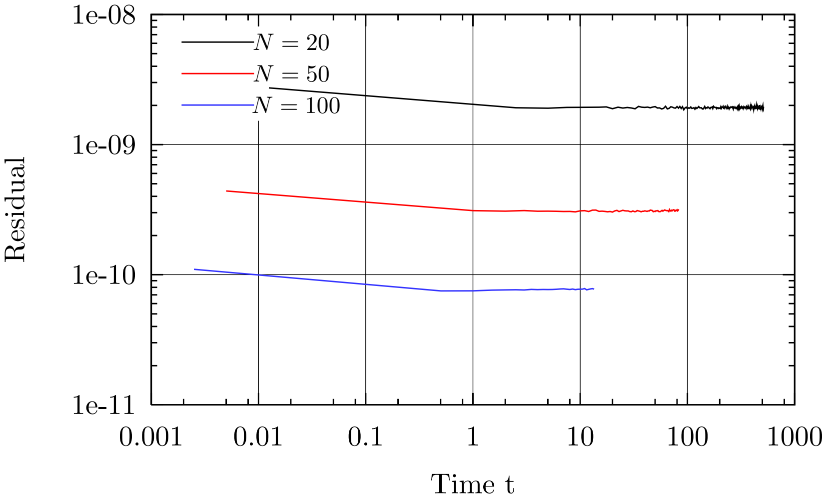


Fig. 3.27: Robust-Stability test in closed universe. QR scheme. Residual (Sum of the absolute value of each Regge equation) for different resolutions versus time is shown.

The LU scheme shows a different performance. With $N = 20$ we have the result shown in Fig.(3.28). The LU scheme yields a linear raise of the residual.

The Test 1 shows that the QR method is better than the LU method in the sense, that the

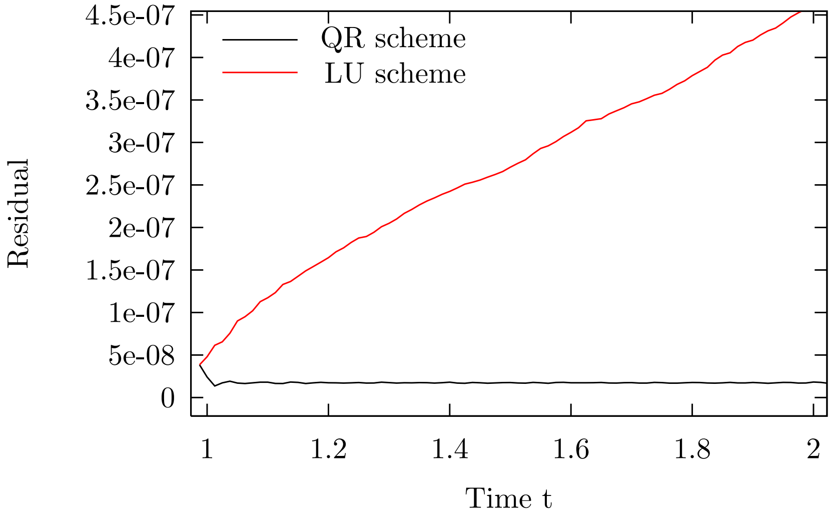


Fig. 3.28: Robust-Stability test, Closed Universe: LU scheme vs QR scheme. Residual (Sum of the absolute value of each Regge equation) is plotted.

residual, here that is the sum of tha absolute value of each Regge equation, is stable with the QR method and raises linearly in case of the LU method. It seems to be sufficient to not constraint the equations corresponding to the shift edges. If the timelike edge is constrainted, qualitatively identical results are obtained. Corresponding plots would lie on top of each other.

Test 2: The Robust-Stability Test in an Open Universe with Unstructured Grids

In this section the Robust-Stability test with the same metric as before is applied to an unstructured grid, which was constructed using the mesh generator `NETGEN`. The use of arbitrary unstructured grids is new to the field of solving Regge equations in Numerical Relativity. With `NETGEN` a unit cube was divided into $5 \times 5 \times 5$ sub-cubes. The triangulations was optimized in `NETGEN` with respect to the volume. Furthermore, the settings very fine and 5 points per edge were used. The main additional effort in this test is to pose right boundary conditions. For this, each edge $e \in \partial\Omega$ was set initially with the unperturbed flat metric. These are the Dirichlet boundary conditions.

As discussed before, we can suppress the evolution of boundary points when we fix additional edges in advance in a way that they fulfill the Regge equations automatically and correspond to the boundary condition already applied to the edges $e \in \partial\Omega$. Therefore it is necessary to set edges $\overline{PA}$ with $P \in \partial\Omega$ and $A \notin \partial\Omega$ which belong to the evolution scheme of a boundary point P. A boundary point has also equation-unknown pairs as inner points have by using the prism method.

The method used here to calculate the necessary edge lengths is the Galassi method. Fig.(3.29) shows the result for the evolution of the residual. After several timesteps of constant residual value the error raises linearly with respect to the time. A timestep of $\Delta t = 0.005$ was applied while the minimum edge length of the unstructured grid has a value of $\Delta x = 0.02$ guaranteeing a Courant factor of $1 : 4$. The QR method does not show any indicator for a collapse of the simulation within the observed time range of $[0, 500]$. Contrary, the LU method crashes within a few timesteps (< 20).

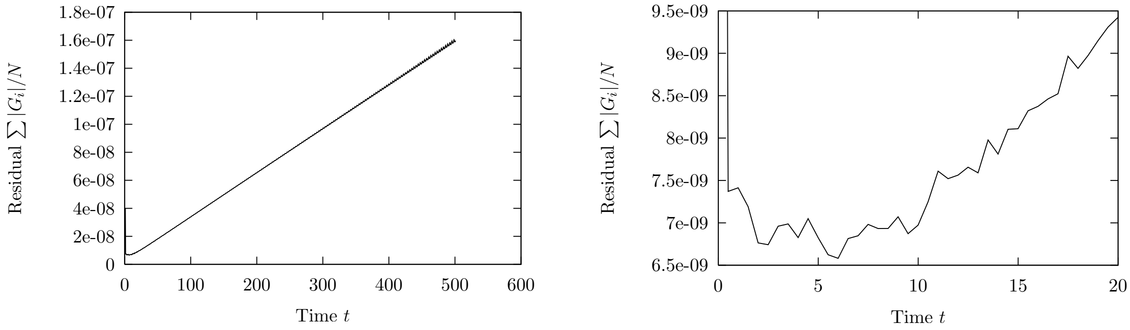


Fig. 3.29: Robust-Stability test in a universe with spatial boundary using the QR scheme and an unstructured grid from the external mesh generator NETGEN. After several time steps of constant residual it raises linearly. A unit cube region was filled with an unstructured grid.

Test 3: The Linear-Wave Test

The next more complicated test for the code from the standard testbeds posed in [AAH$^+$03] concerns a very small gravitational wave on a flat background. The squared amplitude of the wave vanishes within numerical accuracy. The metric is:

$$\begin{aligned} \mathrm{d}s^2 &= -\mathrm{d}t^2 + \mathrm{d}x^2 + (1+b)\mathrm{d}y^2 + (1-b)\mathrm{d}z^2 \\ \text{with } b &= 10^{-8}\sin\left(2\pi(x-t)\right) . \end{aligned} \tag{3.41}$$

It describes a linearized plane wave traveling in x-direction. The setup is the same as for the Robust-Stability test in the closed universe: Again, the domain is extended only in x-direction, $x \in [-0,5,0,5]$ and is triangulated with the approach by Gentle and Miller. Zero-Shift conditions were used as shown in Fig.(3.26). In y- and z-direction the triangulation was extended as small as possible, so that the triangulation does not collapse. The timestep is chosen to be $\Delta t = \Delta x/4$.

Again, the QR scheme yields favorable results which can be expected since the flat background metric is dominant. The behavior of the wave is tracked while crossing its initial state again and again due to the closed character of the universe. From [AAH$^+$03] at least a phase shift can be expected. The value of interest is the deviation of the flat metric perpendicular to the expansion direction. For that, the g_{zz} components were calculated as follows:

$$l_{ij}^2 = \sum g_{\mu\nu}(x_i - x_j)^\mu (x_i - x_j)^\nu \tag{3.42}$$

$$\Rightarrow g_{\mathrm{zz}} = \frac{l_{ij}^2}{(\Delta z)^2} \quad \text{for } \overrightarrow{P_iP_j} \text{ in } z-\text{direction} . \tag{3.43}$$

Here, i and j are point indices. For $g_{\mathrm{zz}}(x = P_i)$ a point P_j is chosen for which $\overrightarrow{P_iP_j}$ has a z-component only. In this way, g_{zz} can be calculated easily. Fig.(3.30) shows the residual with respect to the time. First, the evolution of the residual on a large time scale is shown. Second, a small close-up is chosen to show the oscillations which occur on a very small time scale. Fig.(3.31) shows the phase shifts which occur after a different amount of crossings N. Here, one crossing describes the time range which is needed theoretically to let the wave cross its initial state again due to the character of the closed universe. We see that not only the phase of the wave is changing but also the range of values which the corresponding metric component takes. The mean value of g_{zz} is slightly dropping. This can be an artifact due to the method of extracting the metric component: If the geometry deviates slightly the edge which is initially completely in z-direction turns away and so the extracted metric component gets disturbed.

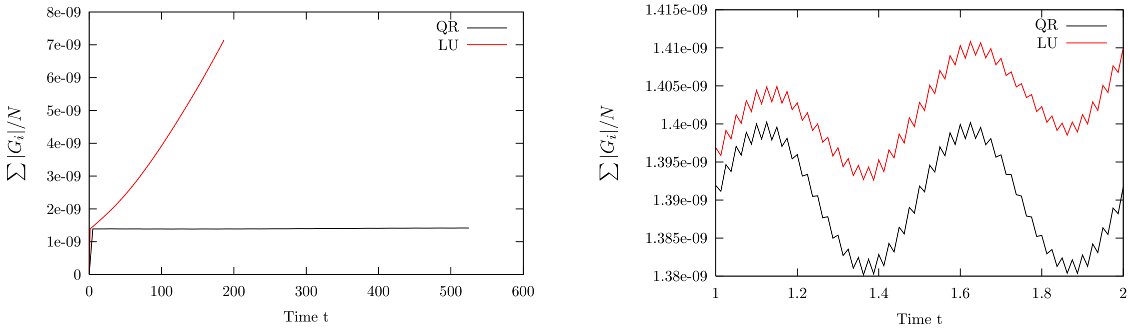


Fig. 3.30: Linear-Wave test. Evolution. Residual is plotted. *Left.* Large scale. QR scheme gives smaller residuals than LU scheme. *Right.* Small scale. High-frequent oscillations occur.

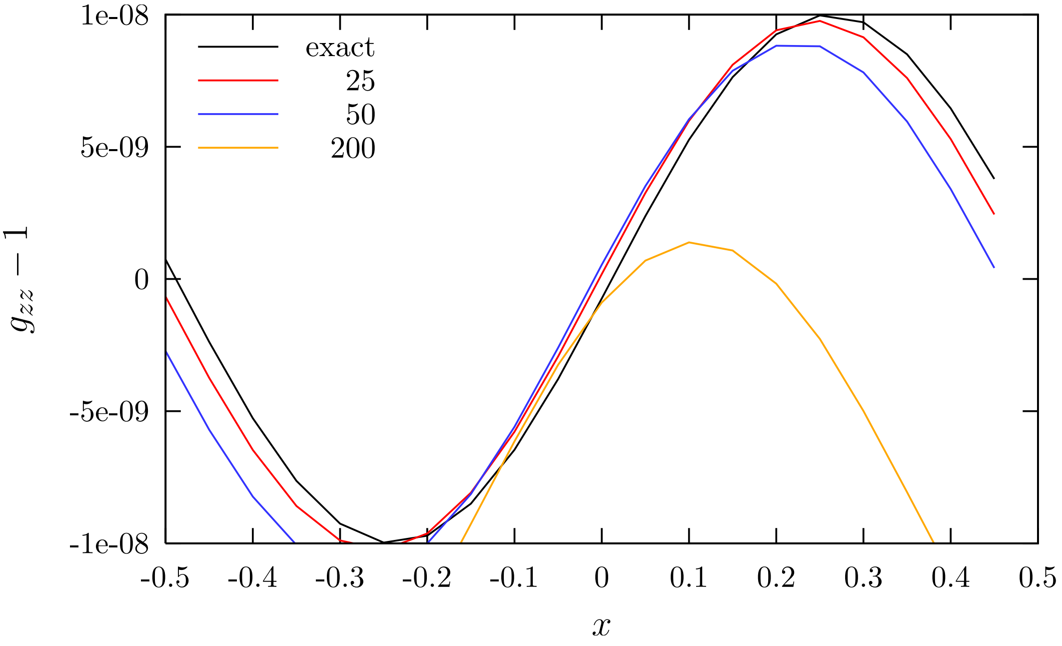


Fig. 3.31: Linear-Wave test. Phase Shift. Number of crossing times is variied. The y-axis shows the deviation from the flat value $g_{\mathrm{zz}} = 1$.

Discussion of Results

From the three tests it can be seen, that the QR scheme is favourable. In all tests this scheme made a long-term Time-Evolution possible.

Considering the Linear-Wave test, the phase shift of the linear wave growths with increasing crossing times as expected from the previous publications [AAH^{+}03] and [BHW^{+}07]. In addition, the corresponding metric components drop slightly. This can be due to a small twist of the geometry which has direct influence to the extraction of metric components. When an edge is not located exactly in the z-direction for instance, then the extracted metric component g_{zz} includes an error which is not directly connected to the Time-Evolution of Regge Calculus itself but does occur due to the extraction method showed in Eq.(3.42).

For the Robust-Stability test on an unstructured domain with spatial boundary the QR scheme makes an evolution just possible. The LU scheme fails in this case almost immediately. Here, the residual raises linearly after a short period of constant value. This can be due to the introduction of the boundary or the unstructured grid only because the Robust-Stability test in a closed spacetime does show a constant residual.

Anticipating the results in the next sections it must be stated that the huge difference between QR scheme and LU scheme vanishes in cases when the metric deviates dominantly from flat spacetime and a structured grid is applied. Especially for the Kasner universe and the Gowdy universe, which were calculated using structured grids, both methods yield same results.

Note on Unperturbed Flat Spacetime

It is not useful to run the code with an unperturbed flat spacetime since the Regge Equations are already fulfilled because the deficit angles are zero within machine precision. In this case the Newton scheme does not start and the solution stays static. Whether the code is useless then in sense of Time-Evolution, it is a good occasion to test the code for correct deficit angle calculation.

3.4.4 Convergence in Regge Calculus

In this section the convergence of Regge Calculus within appropriate error measures are investigated in detail. For that, the Linear-Wave test was chosen. It is introduced in *Test 3* of the former section. All parameters were taken over except for the time step width Δt and the spatial step width $\Delta x = 1/N$ which were changed according to the upcoming results. N denotes the number of cubes in x-direction.

To my best knowledge, at least three papers on Regge Calculus deal with numerical convergence issues. The results are shortly sketched in the following:

[Mil95] Several spacetimes were considered on a structured triangulation of a cube. The squared edge length were calculated to numerical accuracy. The error measure $E = \sum |G_i|$ was used. The index i runs over edges of *one* cube. Therefore, the number of edges is constant with respect to the grid resolution and the error needs not to be normalized. If Δx characterizes the edge length of the cube in each direction, spacelike and timelike, then $E = E(\Delta x^4)$ for Schwarzschild spacetime and Kasner spacetime was investigated No Time-Evolution was applied.

[Gal93] By means of the Kasner universe one point was evolved with the Time-Evolution scheme. Here, the error is defined with $E = \sqrt{\sum {G_i}^2}$. The index i runs over all Regge equations corresponding to one point only. Because the LU method was used here, the sum contains only four terms, namely the constraint equations due to the assumed simplicial Bianchi identity. A triangulation of a cube was taken to decompose the cube into simplices. The timestep Δt had no influence on the error E. With refining the cube edge length Δx in each of the three spatial directions, the error dropped with $E = E(\Delta x)^3$.

[BG01] In this paper the Kasner universe was considered on a structured grid which is expanded in one direction only. The other two spatial dimensions were extended to an amount that the algorithm is allowed to calculate deffict angles and does not collapse. By changing the number of cubes $N \sim 1/\Delta x$ in x-direction and applying a fixed Courant condition of $\Delta t = \Delta x/5$ the relative error of one edge is oscillating. It is shown that the amplitude of this oscillation drops with $E = E(\Delta x)^2$. The oscillation is superposed with a linear raise of the error with respect to time.

We see that convergence was investigated with respect to the residual $\sum |G_i|$ or $\sqrt{\sum {G_i}^2}$ respectively. Alternatively, the relative error of a squared edge length $(l_i^{\mathrm{num}2} - l_i^{\mathrm{an}2})/l_i^{\mathrm{an}2}$ was considered. Since the residual does not measure the error to the analytical solution it drops out as a meaningful measure to compare analytical and numerical solution. On the other hand the residual is very useful to discuss the stability of the numerical scheme. Also the deviation of edge lengths to their analytical expectated value is no good error measure since this deviation depends on the triangulation.

For example, if we consider the sum of the squared edge lengths as a norm on which we measure errors, then the analytical solution should give a constant value for this norm. Of course, it is demanded that the norm of the analytical solution is independent from the triangulation since the exact solution is independent from any discretization scheme. Let us assume the Euclidean metric on $\Omega = [0, 1]$. If we now calculate the sum of the squared edge lengths on the interval $[0, 1]$ with only one edge, we get a norm of the exact solution of $\sum l_i^2 = 1^2 = 1$. If we half the interval, we have two edges of length $0,5$. Now, the norm of the exact solution would be $\sum l_i^2 = (1/2)^2 + (1/2)^2 = 1/2$. The values are not the same. Moreover, in the continuum limit the norm vanishes. This points out that we can not take the squared edge lengths to construct an appropriate norm. This effect gets even worse when we try to assign weights proportional to the surrounding volume V_i of each edge to the squared edge lengths. That means we can also not construct meaningful error measures with norms of kind $\sum V_i l_i^2$. Taking the edge length l_i itself does circumvent this problem only in one dimension. In higher dimensions the summed up edge length depends on the triangulation again and makes the error measure not feasible.

Instead in this work the 3-metric γ_{ij} is used to calculate errors. i and j run over all three spatial indices. Since Regge Calculus can regarded to be a Finite-Element method with piecewise constant basis functions we can also take over the standard error norms from Finite-Element theory. In this work, the infinity norm and the L_2 norm are taken. Each of the six 3-metric components are summed up to give a pointwise defined value. If γ^{num} refers to the 3-metric gained from the numerical simulation and γ^{an} is the analytical 3-metric, then u_{ij} is defined with

$$u_{ij}(\mathbf{x}, t) := \gamma_{ij}^{\mathrm{num}}(\mathbf{x}, t) - \gamma_{ij}^{\mathrm{an}}(\mathbf{x}, t)\,,$$

then the Infinity norm and the L_2 norm are defined with

$$\begin{aligned} ||u(\cdot, t)||_\infty &= \max_{\mathbf{x}\in\Omega_t} \max_{ij} |u_{ij}(\mathbf{x}, t)|\,, \\ ||u(\cdot, t)||_{L2} &= \int_{\Omega_t} \sum_{ij} |u_{ij}(\mathbf{x}, t)|^2 \,\mathrm{d}V \end{aligned} \qquad (3.44)$$

In this notation $u(\cdot, t)$ emphasizes that the error is calculated on a given hypersurface with time coordinate t. Also Ω_t denotes the region of spacetime with fixed t. The vector $\mathbf{x}$ is a

3-vector pointing to a point at the hypersurface of time t. In the applied code vector $\mathbf{x}$ ranges over all nodes of the net and the value of the integral was calculated by means of a weighted sum over all nodes of the simplex. For instance, if $f(\mathbf{x})$ is an arbitrary function or a metric component respectively, then:

$$\int_{\Omega_t} f(\mathbf{x}) \approx \sum_{\text{simplices}} V_s \cdot \sum_{\text{nodes of simplex}} \frac{f(\mathbf{x})}{4} \, .$$

V_s denotes the volume of the 3-simplex. This volume can be calculated with the *Cayley-Menger determinant*:

$$288 {V_s}^2 = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & {l_{12}}^2 & {l_{13}}^2 & {l_{14}}^2 \\ 1 & {l_{21}}^2 & 0 & {l_{23}}^2 & {l_{24}}^2 \\ 1 & {l_{31}}^2 & {l_{32}}^2 & 0 & {l_{34}}^2 \\ 1 & {l_{41}}^2 & {l_{42}}^2 & {l_{43}}^2 & 0 \end{vmatrix}$$

The matrix is symmetric.

The metric components g_{ij} are stated upon a global coordinate system. If we recall the equation of determining squared edge lengths from the metric

$${l_{ij}}^2 = \sum_{mn} \gamma_{mn} (x_i - x_j)^m (x_i - x_j)^n$$

we can also turn this equation to a system of equations to recover the metric components g_{mn}:

$$\sum_{mn} \left[(x_i - x_j)^m (x_i - x_j)^n \right] \cdot \gamma_{mn} = {l_{ij}}^2$$
$$\Rightarrow M_{(ij)(mn)} \cdot \gamma_{(mn)} = (l^2)_{(ij)} \, .$$

Now we can introduce two arbitrary but complete index maps $(ij) \rightarrow a$ and $(mn) \rightarrow b$ to make the formulation of the system of equations most clear:

$$M_{ab} \cdot \gamma_b = (l^2)_a \, .$$

Since a 3-simplex consists of six edges and also six components of the 3-metric γ_{mn} are looked for, this gives us a quadratic system of equations of dimension six which can be solved easily. In the upper equations i and j are indices refering to a point of the triangulation. The indices m and n refer to metric components in the given global coordinate system.

The error measures defined in Eq.(3.44) were now applied in several Linear Wave tests. The spatial discretization length Δx were set to every value in $\{1/10, 1/20, 1/40, 1/80\}$. Also, the timestep was set to several values. If $t_{\text{ref}} = 0.00625$ is the reference timestep, then t/t_{ref} was set to every value in $\{1/2, 1, 2, 4, 8\}$. The reference timestep was set in a way that integer valued Courant ratios are guaranteed. The table Tab.(3.5) gives an overlook over the corresponding Courant factors $\Delta x / \Delta t$. The results for both norms are noted in table Tab.(3.6). The entries marked with the symbol $\bigotimes$ indicate that the numerical simulation did break down within one single timestep. This means in each run a signature error occured after the Time-Evolution of a single point in the initial hypersurface. These breakdowns correspond to a Courant condition. All simulations

t/t_{ref}	N=80	N=40	N=20	N=10
8	0.25	0.5	1	2
4	0.5	1	2	4
2	1	2	4	8
1	2	4	8	16
1/2	4	8	16	32

Tab. 3.5: Convergence of Linear-Wave test. To timestep and spatial step corresponding Courant factors.

t/t_{ref}	N=80	N=40	N=20	N=10
8	⊗	⊗	⊗	⊗
4	⊗	⊗	⊗	32.8
2	⊗	⊗	16.9	32.0
1	⊗	8.47	16.2	31.5
1/2	4.25	8.14	15.9	31.3

t/t_{ref}	N=80	N=40	N=20	N=10
8	⊗	⊗	⊗	⊗
4	⊗	⊗	⊗	14.93
2	⊗	⊗	7.45	14.84
1	⊗	3.72	7.42	14.82
1/2	1.86	3.71	7.41	14.80

Tab. 3.6: Convergence of Linear-Wave test. *Top.* Infinity Norm $||u(\cdot,t)||_\infty$ after one crossing of the linear wave. *Bottom.* L_2 Norm $||u(\cdot,t)||_{L2}$ at maximum during first crossing. All values are noted in terms of 10^{-10} of the norm of the exact solution.

with a factor of 1 : 2 or less crash, while all simulations with a factor of 1 : 4 or better run successfully. A more precise test for $N = 40$ did show that the simulation does not break down if the Courant factor is slightly better than 1 : 2. The parameter region from 1 : 4 to 1 : 2 were covered with several test runs and a ratio of 1 : 1.203 yield reasonable results.

Furthermore, from Tab.(3.6) we see that the error drops linearly with the step size Δx while the error is almost constant with varying timestep Δt. This effect is contributed to the discretization error of the metric. Since the pointwise defined metric becomes a piecewise flat metric by using Regge Calculus, nautrally a discretization error occurs at the starting time already. This is analogous to approximate an integral with a sum of rectangles for instance. Furthermore we see that this error dominates the error evolution in time direction. Fig.(3.32) shows clearly that the discretization error is much higher than the error growth within time in both norms. In this figure a Courant factor of 1 : 4 was chosen. Therefore timestep Δt and spatial step size Δx change simultaneously. If we substract the discretization error $||\gamma^{\mathrm{an}}(\cdot,0) - \gamma^{\mathrm{num}}(\cdot,0)||$ we are able to investigate the convergence

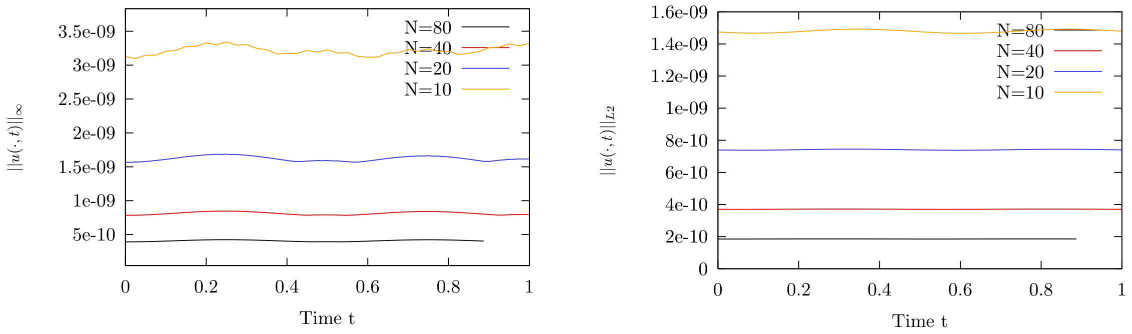


Fig. 3.32: Linear-Wave test. Evolution. Error norms are plotted in terms of the norm of the exact metric. The Courant factor is set to 1 : 4.*Left.* Infinity norm $||u(\cdot,t)||_\infty$ is plotted versus time. *Right.* L_2 norm $||u(\cdot,t)||_{L2}$ is shown.

order in time. Fig.(3.33) shows the corresponding graphs for both error norms. Now, the order of convergence can be estimated with the graphs. The Infinity norm shows an oscillation superposed with a linear raise of the error. By counting grid lines a convergence order of two can be detected: With doubling the grid resolution the error drops by a factor of 1/4. The L_2 norm of the error shows a much stronger oscillation. But the order of convergence can be estimated with two as well when we look at the maximas of the oscillations.

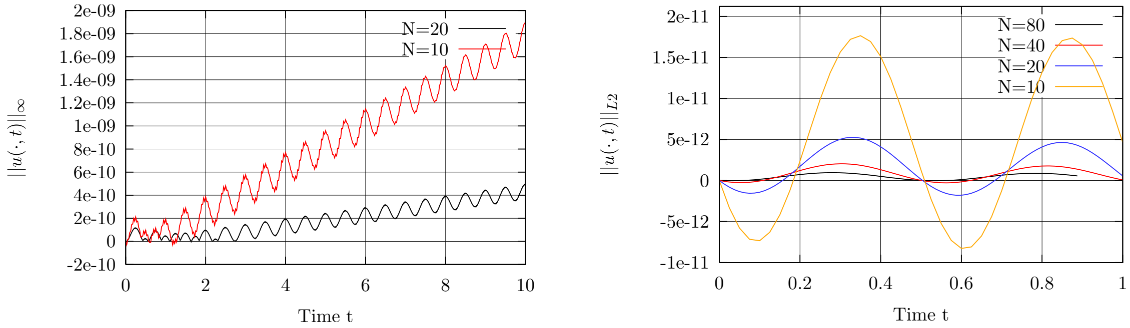


Fig. 3.33: Linear-Wave test. Evolution. Error norms are plotted in terms of the norm of the exact metric. The discretization error for the initial hypersurface is substracted. The Courant factor is set to 1 : 4. *Left.* Infinity norm $||u(\cdot,t)||_\infty$ is plotted versus time. *Right.* L_2 norm $||u(\cdot,t)||_{L2}$ is shown.

Upto now we have investigated the error with a constant Courant factor of 1 : 4. By varying the step size Δx only we get error graphs as shown in Fig.(3.34). There, the time step is constantly set to $t = 0.5\, t_{\text{ref}}$. In Fig.(3.35) the timestep is variied only and the spatial step width is constant at $\Delta x = 1/40$. In the last case the discretization error is not substracted since it is the same for all different timesteps.

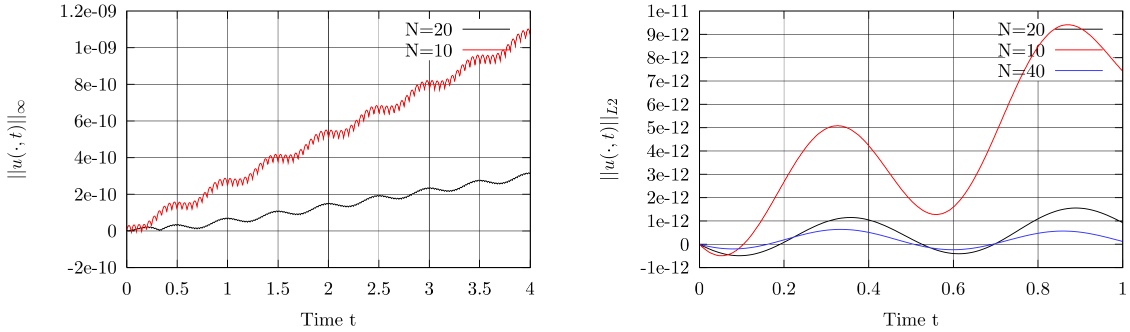


Fig. 3.34: Linear-Wave test. Evolution. Error norms are plotted in terms of the norm of the exact metric. The discretization error for the initial hypersurface is substracted. The time step width Δt is set to $1/2 \cdot t_{\text{ref}}$. *Left.* Infinity norm $||u(\cdot,t)||_\infty$ is plotted versus time. *Right.* L_2 norm $||u(\cdot,t)||_{L2}$ is shown.

Along the variation of the spatial step size Δx at fixed Δt, we see that the infinity norm is dropping again proportional to a convergence order of two. The L_2 norm is oscillating. A convergence order of two can be estimated if consdidering the maximas. Moreover it seems that the difference between the plot of $N = 10$ and $N = 20$ is of even higher order. This can be attributed to the very coarse grid of $N = 10$ which has not the resolution to cover maxima, minima and zeros of the linear wave in contrast to a resolution of $N = 20$.

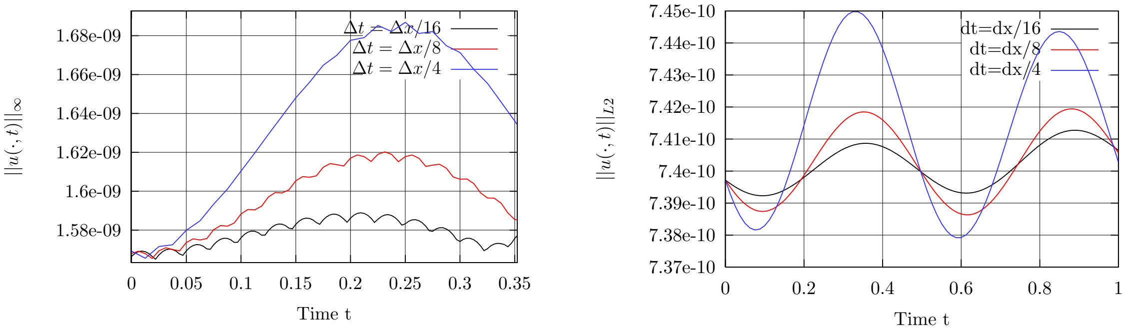


Fig. 3.35: Linear-Wave test. Evolution. Error norms are plotted in terms of the norm of the exact metric. The spatial step width Δx is set to $1/40$. *Left.* Infinity norm $||u(\cdot,t)||_\infty$ is plotted versus time. *Right.* L_2 norm $||u(\cdot,t)||_{L2}$ is shown.

The infinity norm with constant Δx and variing Δt can be estimtated with a linear convergence order when we look at the maximas of the oscillation. The L_2-norm is oscillating strongly but a linear convergence order can be estimated when we consider the maximas only. If the Courant factor is too high then the time step size gets much smaller than the spatial step size Δx. Therefore the error can be almost fully attributed to the discretization in spacelike direction and the benefit of a smaller timestep vanishes.

Summarizing all investigations we can state that first the error drops linearly with Δx when we consider the error to the exact solution γ_{ij}^{an}. When we substract the discretization error and watch how the error evolves in time we see a quadratic convergence in Δx and also a quadratic convergence in Δt when we apply a constant Courant factor:

$$||u(\cdot,t)||_\infty - ||u(\cdot,0)||_\infty = \mathcal{O}(\Delta x)^2 + \mathcal{O}(\Delta t)^2$$
$$||u(\cdot,t)||_{L2} - ||u(\cdot,0)||_{L2} = \mathcal{O}(\Delta x)^2 + \mathcal{O}(\Delta t)^2$$

If we take a fixed spatial step size then the error drops only linearly in Δt. This can be attributed to a non-constant Courant ratio and would point out a restriction on the generation of the triangulation. Also the tracked point coordinates are changing and it is not perfectly clear at which coordinates the new evolved point is located. As shown before, the Galassi method and the Zero-Shift conditions are exact only when we deal with flat hypersurfaces. In the Linear-Wave test the wave travels along the x-direction and in this direction the metric changes continuously. That means the applied Zero-Shift condition along the x-axis is not exact in contrast to the conditions applied in y- and z-direction where the metric components are constant.

Normally, from a Finite-Element method with piecewise constant basis functions, as Regge Calculus can be interpreted, we would expect the error to decrease only linearly. The quadratic convergence with respect to $||u(\cdot,t)||_\infty - ||u(\cdot,0)||_\infty$ and $||u(\cdot,t)||_{L2} - ||u(\cdot,0)||_{L2}$ can be attributed to a discontinuous Galerkin method and it would be from great interest to relate Regge Calculus to such a method.

At last it was investigated that the Courant ratio in the Linear-Wave test match exactly the theoretical derivation from section 3.2.3. This means that only causality restricts the timestep.

3.4.5 Dominantly Non-linear Closed Spacetimes

In this section dominantly non-linear closed spacetimes are considered. That means that this section includes metrics which non-linear terms are not of magnitudes smaller than the flat terms. The first test considers the Kasner universe and has a closer look on how prescise the calculated metric components match the analytical ones. The second test evolves the Gowdy Spacetime. It is much more complicated than the first test since the metric components in spatial directions evolve exponentially and linear respectively.

Test 1: The Kasner Universe

The Kasner universe is described with the metric

$$\mathrm{d}s^2 = -\mathrm{d}t^2 + t^{2p_1}\mathrm{d}x^2 + t^{2p_2}\mathrm{d}y^2 + t^{2p_3}\mathrm{d}z^2\,. \tag{3.45}$$

It describes an expanding universe which hypersurfaces $t = \text{const.}$ are flat. Only when comparing the distance between two points in a hypersurface at two different times, it becomes apparent that the spacetime is expanding. This property guarantees that the Zero-Shift conditions are exact. The lapse edge not only emanates perpendicular to one special 3-simplex which was selected to fixing lapse and shift but it also emanates perpendicular to each other 3-simplex in the hypersurface.

The Kasner universe was first calculated with Regge Calculus by Gentle and Miller in [GM97]. In the present work, a lot of parameters from this work were kept over. That is the triangulation, the constraint conditions and the parameters p_i: $p_1 = p_2 = 2/3$ and $p_3 = -1/3$. This makes a comparison possible. A grid resolution of $6 \times 6 \times 6$ was used. During calculation it turned out that in this special case the LU scheme and the QR scheme yield same results. The grid distance was chosen with $\Delta x = 0,005$ and $\Delta t = \Delta x/4$. These are parameters almost equivalent to [GM97]. The evolution of edge lengths in x- and z-direction are shown in Fig.(3.36). They show the expected qualitatively behaviour. This result is equivalent to the one obtained in [GM97].

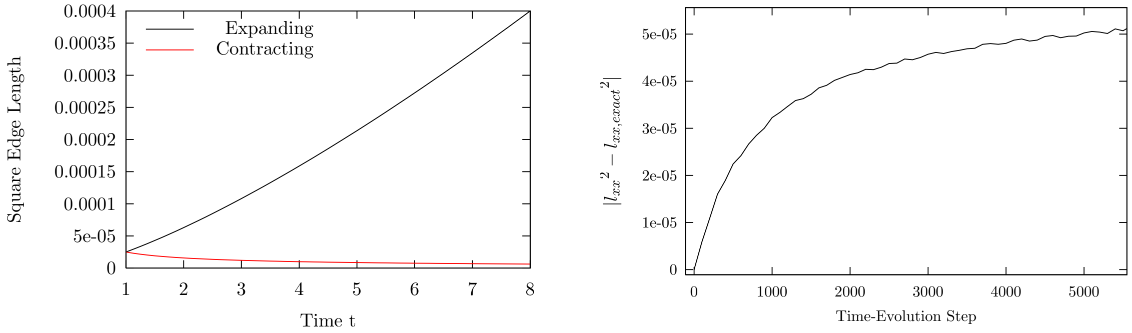


Fig. 3.36: Kasner universe. Evolution of squared edge lengths. *Left.* The evolution of calculated and analytic squared edge lengths. The error is within line width. *Right.* The relative error of the l_{xx} edge.

From the Kasner metric the analytical expected squared edge length in x-direction can be easily derived with the ususal equation:

$$l_{ij}^2 = \sum_{mn} g_{mn} \cdot (x_i - x_j)^m (x_i - x_j)^n \,,$$
$$\Rightarrow l_{xx}^2 = g_{xx} \cdot (x_1 - x_2)^2 \,.$$

The index $(ij) = (xx)$ means that the edge is aligned in x-direction. Therefore only coordinate differences in x-direction are not zero and the sum reduces to one single term. x_1 and x_2 denote the x-coordinate of the first and second point of the line respectively. The metric index $(mn) = (xx)$ points to the corresponding metric element.

With the analytical value for each squared edge length the quantitative deviation from the numerical value was investigated next. For that, the relative error of the averaged edge length in x-direction is shown. A first comparison of numerical and analytical squared edge lengths gives us the result that the error is within linewidth. Then, the relative error is plotted on the right hand side of Fig.(3.36). It shows a raising error which is decending. We see that the magnitude of the relative error is around 10^{-5}.

The residual of the Regge equations expressed by the sum of the absolute value of each equation is plotted in Fig.(3.37).

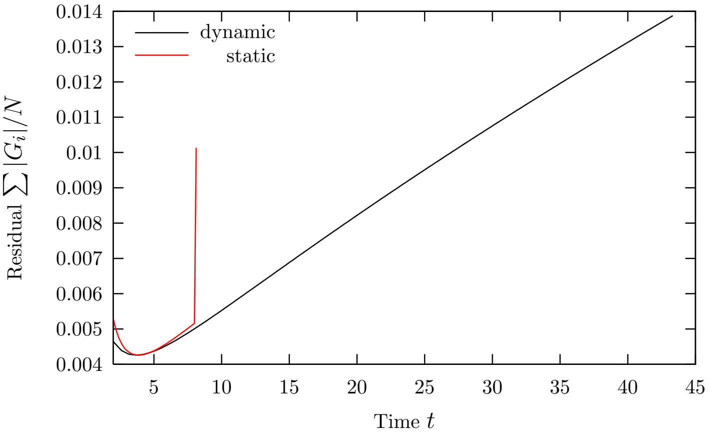


Fig. 3.37: Kasner universe. The residual is plotted. Static lapse let crash the code. With dynamic lapse the code do not crash.

By the evolution of the residual an interesting fact can be observed. With fixing the squared edge length of the lapse edge for all times to the amount corresponding to $\Delta t = \Delta z(t = 1)/4 \Rightarrow l_{\mathrm{tt}}^2 = l_{\mathrm{zz}}^2(t = 1)/16$ the evolution crashes at $t \approx 8$. This can be expected, since the ratio of timelike edge to the minimal spacelike edge is increasing in time and at some point a Courant condition is violated. To determine the critical ratio for the Kasner universe, the following estimate can be derived using the squared edge length in z-direction. Let $l_{zz}^2(t = 1)$ denote the initial squared edge length along cube edges in the spacelike hypersurface at initial time $t = 1$, then:

$$
\begin{aligned}
{l_{\mathrm{zz}}}^2(t=8) &= 8^{-2/3} \cdot {l_{\mathrm{zz}}}^2(t=1) \\
&= \frac{1}{4} \cdot {l_{\mathrm{zz}}}^2(t=1) \\
\Rightarrow \quad -l_{\mathrm{tt}}^2 &= \frac{1}{16} \cdot {l_{\mathrm{zz}}}^2(t=1) \\
&= \frac{1}{4} {l_{\mathrm{zz}}}^2(t=8) \\
\Rightarrow \quad \frac{\Delta t}{\Delta x} &= \sqrt{\frac{{l_{\mathrm{zz}}}^2(t=8)}{-l_{\mathrm{tt}}^2}} \\
&= \frac{1}{2}\,.
\end{aligned}
$$

Therefore the critical Courant ratio is 1 : 2. This conincides with the Courant ratio observed with the Linear-Wave test and gives greater confidence into the theoretical derivation. Because here we started with a ratio of 1 : 4. This ratio gots worse with increasing time and the numerical scheme did break down at a ratio of exactly 1 : 2.

In a succeeding test an alternative equation for the lapse edge was stated to let the evolution run more far. The lapse edge was initialized with a squared edge length of

$$
{l_{\mathrm{tt}}}^2(t,\mathbf{x}) = -\frac{1}{16} {l_{\mathrm{zz}}}^2(t,\mathbf{x})\,,
$$

The dependency on time t and $\mathbf{x}$ points out that the lapse is not static anymore and is not constant in time and space but is calculated locally for each point in a hypersurface and for each new timestep. ${l_{\mathrm{tt}}}^2$ denotes the squared edge length of the lapse edge. ${l_{\mathrm{zz}}}^2$ denotes the squared edge length in z-direction. With this condition the test yields the result on the right hand side in Fig.(3.37) labeled as *dynamic*. In this run the time t corresponds to the time coordinate defined in Eq.(3.45). That means it was taken care to map the time step number to the right time t. The run shows a stable evolution. The residual is dropping first and then raises slightly. This result seems prior to [GM97]. There, the Regge equation of the timelike edge alone (called Hamiltonian constraint there) is raising by a level of two magnitudes in a time range from $t = 1$ to $t = 10$. This can be derived from the energy measure in [GM97]. There, *energy* is called the sum of constraint equations, that are the constraint equations in the LU scheme. In the LU scheme all other but the constraint equations are solved exactly within numerical accuracy. So, the sum of constraint equations (energy) can be replaced with the sum of all equations (residual). When we now compare this energy measure with the residual of the present work we see that the error raise of this work is lower. When comparing absolute values from the present work with these from [GM97] they differ. This is due to a scaling factor in the Regge equations which does not influence the result.

Summarizing the Kasner test, it can be stated that in this case, where squared edge lengths are changing and these dynamics are reproduced within a relative error of magnitude 10^{-5}, Time-Evolution with Regge Calculus is a success and proves feasibility in a non-linear regime.

Test 2: The Gowdy Wave Test

The Gowdy metric, introduced by Robert H. Gowdy, 1971 in [Gow71], describes a closed universe with gravitational waves inside. The metric reads:

$$\mathrm{d}s^2 = t^{-1/2}e^{\lambda/2} \cdot (-\mathrm{d}t^2 + \mathrm{d}z^2) + t\,\mathrm{d}w^2 \text{ with} \tag{3.46}$$
$$\mathrm{d}w^2 = e^{P}\,\mathrm{d}x^2 + e^{-P}\,\mathrm{d}y^2\,. \tag{3.47}$$

P and λ are functions of t and z only. Also, per assumption, they must be periodic in z-direction. These functions are ambiguous and must be determined uniquely by restricting to a special case. The derivation is shortly sketched in [AAH$^+$03]. With the metric applied to the Einstein evolution equations we get

$$\frac{\partial^2}{\partial t^2}P(t,z) + \frac{1}{t}\frac{\partial}{\partial t}P(t,z) - \frac{\partial^2}{\partial z^2}P(t,z) = 0\,. \tag{3.48}$$

The constraint equations become,

$$\frac{\partial}{\partial t}\lambda(t,z) = t((\frac{\partial}{\partial t}P(t,z))^2 + (\frac{\partial}{\partial z}P(t,z))^2\,,$$
$$\frac{\partial}{\partial z}\lambda(t,z) = 2t\frac{\partial}{\partial z}P(t,z)\frac{\partial}{\partial t}P(t,z))\,.$$

The general solution of Eq.(3.48) is a linear combination of terms of the kind $\alpha \log t + \beta$ and terms $Z_0(2\pi n t)\cos(2\pi n z)$, with α and β being real constants, $n \in \mathbb{Z}$ and Z_0 being a linear combination of Bessel functions J_0 and Y_0. Now, we choose a special solution with

$$P(t,z) = J_0(2\pi t)\cos(2\pi z)\,.$$

With this special solution the function λ becomes

$$\lambda(t,z) = -2\pi t\, J_0(2\pi t)J_1(2\pi t)\cos^2(2\pi z) + 2\pi^2 t^2 \left({J_0}^2(2\pi t) + {J_1}^2(2\pi t)\right) -$$
$$- \frac{1}{2}\left((2\pi)^2\left({J_0}^2(2\pi) + {J_1}^2(2\pi)\right) - 2\pi J_0(2\pi)J_1(2\pi)\right)\,.$$

Here, J_0 and J_1 are Bessel functions.

The metric is expanding exponentially in z-direction because λ is raising proportional to time t. Contrary, the metric in y- and z-direction is raising linearly due to the factor t. The function P oscillates in $[-1, 1]$ only. The time t is chosen to increase from $t = 1$, so an expanding closed universe is modeled. The lapse edge was chosen to be constant for one whole time step: This constant is calculated by taking the minimal spatial squared edge length from the previous hypersurface divided by a factor of -16. This is the same approach as applied in the Kasner universe.

Now, the Regge equations are solved again by means of the triangulation idea for a closed universe of GENTLE and MILLER. The triangulation was extended in z-direction to cover the interval $[-0,5,0,5]$. In x- and y-direction the triangulation is extended minimally, so the triangulation does not crash. The initial triangulation was constructed with analytic values. QR scheme and LU scheme yield again same results as observed in the Kasner universe. In z-direction, $N = 20$ and $N = 50$ cubes were used to cover the domain corresponding to Fig.(3.26). In both cases the evolution crashes suddenly at the same time $t \approx 4,1$. According to the graphs in Fig.(3.38) it can almost not be expected before, that the code is going to crash. Only a small oscillation in the squared edge lengths aligned in z-direction can be noticed. Note, the squared edge lengths are proportional to the corresponding metric components if aligned edges can be selected. This is the case with the used structured grid.

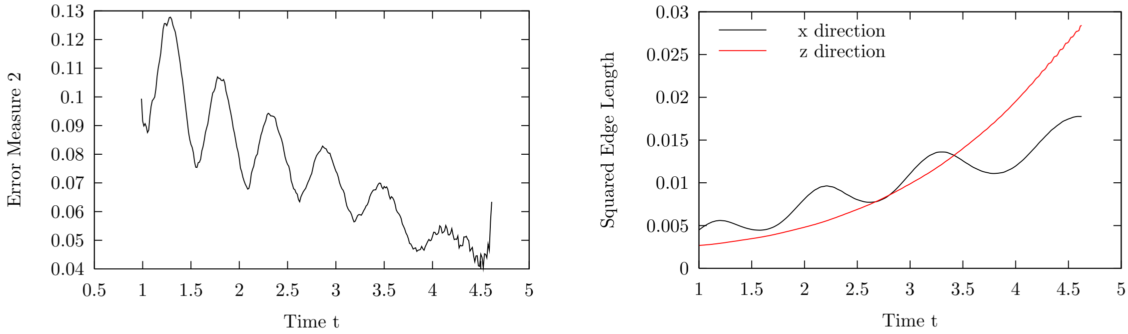


Fig. 3.38: Gowdy Wave Evolution. *Left.* The Sum of Regge equations (Error Measure 2) is shown. It shows an oscillating decay with a sudden breakdown of the numerical scheme. *Right.* The squared edge lengths (proportional to metric components) show expected behavior with a small oscillation short before the breakdown.

Discussion of Results

The test of the Kasner universe did show, that the analytical edge length and the numerical calculated ones match within a relative error of around 10^{-5}. Furthermore, by choosing a Courant condition with a ratio of timelike edge length to spacelike edge length of at least slightly better than $1 : 2$ the evolution did run arbitrary long. Without guaranteeing this Courant condition the evolution did break down suddenly. This conincides with the result of the Linear-Wave test and the theoretical derivation in section 3.2.3. This result points out that the Courant condition depends on causality only and holds true also for non-linear spacetimes.

The squared edge lengths of the Gowdy test match qualitatively the expected analytical result. In both tests, the QR scheme and the LU scheme yield identical results. In contrast to almost flat spacetimes it can be argued that the LU scheme becomes stable when the non-linearity is significant. At least for the applied structured grids this holds true.

The evolution of the Gowdy spacetime crashes suddenly at a special time $t \approx 4.1$. This showed to be independent from the chosen Courant condition. This is an indicator for an additional restriction on edge ratios or angles respectively.

3.4.6 Static Black Hole on an Unstructured Grid

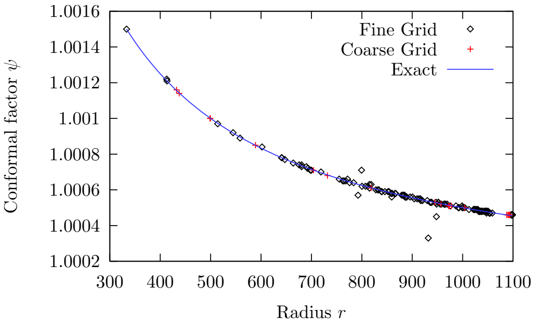
To conclude the result section, the metric given by a static black hole in vacuum is investigated on an unstructured grid. As already discussed in section 2.4.1 and section 3.4.2 the Schwarzschild spacetime in isotropical coordinates is defined as:

$$\mathrm{d}s^2 = -\left(\frac{1-1/2r}{1+1/2r}\right)\mathrm{d}t^2 + \left(1+\frac{1}{2r}\right)^4 (\mathrm{d}r^2 + r^2\,\mathrm{d}\theta^2 + r^2\sin^2\theta\,\mathrm{d}\phi^2)\,.$$

In this equation the mass M is already set to $M = 1$. A ball B_R, where R denotes the radius of the ball, is triangulated with the mesh generator `NETGEN`. This three-dimensional initial grid was then evolved twice with the prism method to get the four-dimensional triangulation. A radius of $R = 1000$ was chosen. The Galassi method was applied to the spatial boundary points. In contrast to the excised domain $B_R \setminus B_a$ from chapter two no ball is excised here from the triangulation. Instead, each point with $r < c$ is handled as a boundary point and c is chosen the smallest possible but to guarantee that all points belonging to simplices that cover

B_a are set to boundary points. These inner points are handled with the Galassi method as well.

With the grid resolution parameter set to *moderate* in `NETGEN` a mesh $\mathcal{T}_0$ was generated on B_R. This grid was refined uniformly once yielding a new triangulation $\mathcal{T}_1$. If the grid $\mathcal{T}_1$ is refined again we get the grid $\mathcal{T}_2$. A comparison of evolutions with $\mathcal{T}_1$ and $\mathcal{T}_2$ shows that an instability located near to the outer boundary evolves faster when we use the finer grid $\mathcal{T}_2$. To see that the conformal factor is plotted versus the radius coordinate direction in Fig.(3.39). From different results of the conformal factor it can be observed that the error begins to raise significantly near to the outer boundary. This instability can either be attributed to an insuffciently stated boundary condition for the outer points or a bad element in the four-dimensional triangulation. Contrary, the discretization error shows the expected behaviour. As with the Linear Wave test the error drops in linear order with respect to the spatial discretization length Δx which is directly related to the number of uniform refinement steps: $\Delta x \propto 1/2^i$. Here, i is the number of refinement steps. Since each edge is halved within one refinement step this relation is obvious. The discretization error in L_2 norm for a different number of uniform refinements is summarized in Fig.(3.39) on the right hand side.



i	$\|\|u(\cdot,0)\|\|_{L_2}$	Convergence Rate
0	$1,53 \cdot 10^{-2}$	
1	$6,99 \cdot 10^{-3}$	$1,13$
2	$3,40 \cdot 10^{-3}$	$1,04$

Fig. 3.39: Schwarzschild Problem. *Left.* Conformal factor is plotted versus radial coordinate r for two different grid resolutions at time $t = 5$. The deviations from the analytical solution is stronger with the fine grid $\mathcal{T}_2$. *Right.* Discretization error $||u(\cdot,0)||_{L_2}$ is shown after a different number of refinement steps i. The error is shown in units of the L_2 norm of the exact solution. The convergence order is calculated corresponding to the error of the actual row and the row above.

The grid $\mathcal{T}_1$ has a higher maximum angle and a lower minimum angle respectively compared to the grid $\mathcal{T}_0$ which can be attributed to the projection of new generated points onto the outer sphere. By applying a different number of optimization steps in `NETGEN` four grids were generated. Fig.(3.40) shows a plane cut at point $(0|0|0)$ with the normal pointing in z-direction. We can see that the triangulation becomes more regular when applying more optimization steps. The color indicates the local error $||u(\mathbf{x},t)||_\infty$ for each node from the triangulation. That means the maximum error from the neighbour simplices is assigned to each node. Then, the visualization software *Paraview* applied a linear interpolation between the node values. For the first two grids the error at breakdown time is shown. For the last two grids the error at $t = 500M$ is drawn. Maximum angle, minimum angle, number of optimization steps and breakdown time is summarized in the caption of figure Fig.(3.40). The corresponding residiual value and the error in the L_2 norm is shown in Fig.(3.41). These results show that with better angles, that means with a higher minimum angle and a lower maximum angle as well, the breakdown time of the evolution increases. For the last two grids the evolutions did not break down till a time of $t = 500M$ but an increasing error in the L_2 norm can already be observed.

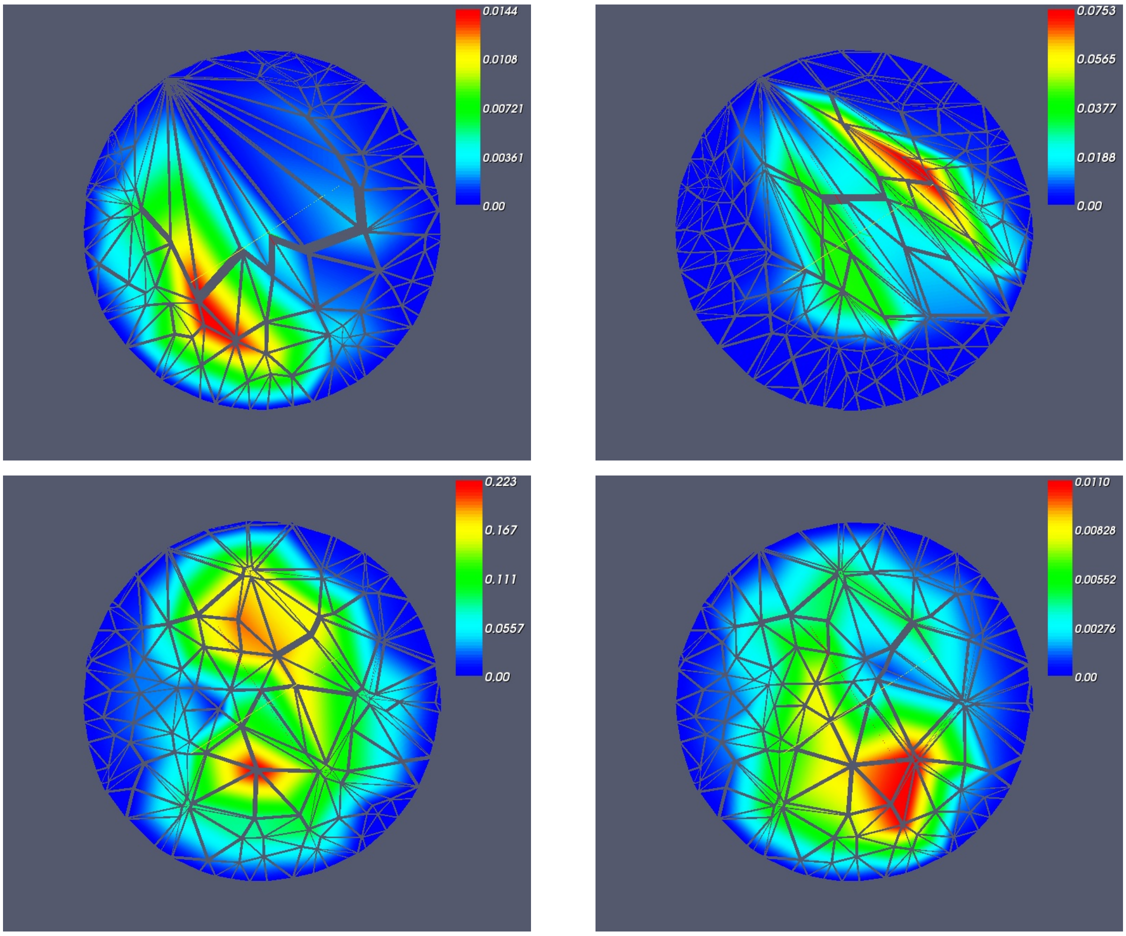


Fig. 3.40: Unstructured Grids for Schwarzschild Evolution. Local error $||u(\cdot,t)||$ is shown at each node. Linear interpolation by visualization program. OS = optimization steps. $\mathcal{T}_1$ used. Minimum and maximum angles in notation: [min,max]. *Top Left.* OS=0. $[9^\circ, 159^\circ]$. $t = 2.9$. *Top Right.* OS=1. $[11^\circ, 151^\circ]$. $t = 5.1$. *Bottom Left.* OS=2. $[24^\circ, 119^\circ]$. $t = 500$. *Bottom Right.* OS=5. $[26^\circ, 113^\circ]$. $t = 500$.

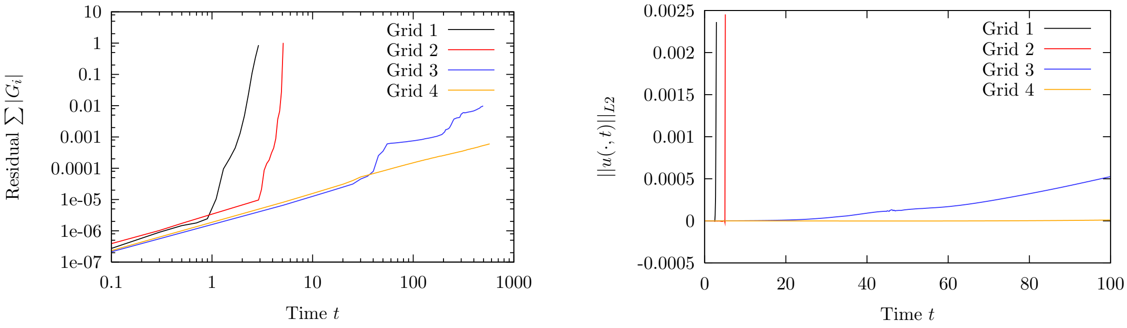


Fig. 3.41: Schwarzschild Problem. Residual and L_2 Error. Grid 1-4 corresponds to optimization steps $0, 1, 2, 5$. *Left.* Residual is plotted versus time. *Right.* L_2 error is plotted versus time.

CHAPTER 4

Summary and Outlook

4.1 Summary

In the first part of the present work it was shown, that the Finite-Element method can be used to solve the initial data problem posed by Bowen and York in [BJ80].

The already developed code `KASKADE` which solves linear elliptical problems only was modified in this thesis to calculate also semilinear differential equations. With linear basis functions applied on a simplicial triangulation initial data for relevant astrophysical problems were successfully computed. The problems of one static black hole, one black hole with linear momentum and the binary black hole problem in a head-on collision were treated. Two different triangulations were considered. First, the external mesh generator `NETGEN` was employed and second, a manually developed mesh was used. The latter preserves the symmetry of the problem to a certain amount. Furthermore adaptive and uniform refinement techniques were compared. The results show that an own constructed grid is preferable but it is reasonable to apply external mesh generation software since the error is only slightly higher in this case. It was shown that not only discretization schemes like finite differences or spectral methods can be applied to problems in Numerical Relativity but also the Finite-Element method works successfully. The Finite-Element method needs not to be developed again for each special problem but a modified external code can be applied successfully.

In the second part, Regge Calculus was used to solve special problems of Numerical Relativity in the Time-Evolution scheme presented by Rafael Sorkin in [Sor75].

First, different triangulation schemes were investigated. As an outcome of this investigation it was shown that not every triangulation of a four-dimensional domain is compatible with the Time-Evolution scheme. An algorithm is presented which evolves arbitrary three-dimensional grids to a four-dimensional triangulation by means of the prism method. These triangulations are always compatible with the Time-Evolution scheme. Closed universes can be successfully handled with the structured triangulation used by Gentle and Miller, published in [GM97]. Non-closed universes give rise to spatial boundaries and boundary conditions. For the first time, the Time-Evolution scheme at the boundary is discussed in this thesis and methods to pose boundary conditions were presented. Not only edges at the boundary but also edges emanating from the boundary must be initialized before the Time-Evolution scheme can be applied. This is shown by means of the Schwarzschild spacetime.

Second, after a presuming test of the Time-Evolution scheme on a 2-sphere, it was shown for the first time that the QR scheme yield longterm stable results for almost flat spacetimes as tested with the Robust-Stability test and the Linear-Wave test. The QR scheme is prior to the LU scheme which takes a simplicial Bianchi identity into account. While the error raises linearly in the LU scheme, the error of the QR scheme is constant. With the Linear-Wave test a phase shift was observed as expected from previous work. Furthermore, the Robust-Stability test on a domain with boundaries was tested, applying the new developed techniques to treat the boundary. Here, the LU scheme crashed immediately while the QR scheme yield long term stable results with a linear increase of the error.

Third, convergence was investigated in detail by means of the Linear-Wave test. Integral norms from Finite-Element theory were applied to the metric which is reconstructed by a fundamental relationship between squared edge lengths and metric components. This approach is completely new to Regge Calculus. Neither integral norms nor the metric itself was ever taken to discuss convergence. As an further outcome an analytical expected Courant condition was verified. It states that the ratio of timelike edge to minimal spacelike edge length must be smaller than $1 : \sqrt{2}$.

Fourth, significantly non-linear spacetimes were investigated. Lapse and shift were applied to improve the evolution. Squared edge lengths of the Kasner universe did match the analytical expected values with a relative error of the magnitude of 10^{-5}. The observed metric components did show the expected behaviour in both cases. With an adjusted lapse condition it was possible to circumvent a breakdown of the simulation in the case of the Kasner universe. The Courant condition which was derived within this thesis could be verified in this case too.

At last, the new developed techniques to treat unstructured three-dimensional grids and boundaries of the domain as well were applied to Schwarzschild spacetime. Several simulations with differently optimized grids were compared. It could be investigated that with better angle properties of the grid the simulation run more stable. A non-optimized grid crashed almost immediately while an sufficiently optimized grid ran stable until $t \approx 500M$.

In this thesis it was shown that the presented simplicial methods are applicable to relevant problems of General Relativity. An already existing code was modified in this thesis to solve the initial value problem of one or two black holes. The Finite-Element method showed robustness to different grids and refinement techniques. The selfmade structured grid `MESHGEN` was slightly preferable to an unstructured grid from the external mesh generation software `NETGEN`. The adaptive refinement technique was prior to the uniform one.

With Regge Calculus, it was possible to calculate many spacetimes. The newly developed techniques concerning QR method, boundary treatment and the implementation of unstructured grids made the simulation of almost flat spacetimes and Schwarzschild spacetime just possible. With this a detailed discussion of convergence, accuracy and Courant condition was done.

4.2 Outlook

In the broad field of Regge Calculus a lot of questions are open and a lot of more investigation is needed to make Regge Calculus a sophisticated method to calculate complex problems. For work which take the new insights of the present thesis further it may be important to identify an angle condition of the used triangulation which gurantees stable evolutions for unstructured grids. This condition could then be preserved by appropriate lapse and shift conditions.

In this work, a $3 + 1$ decomposition was used. A completely different approach would be to solve the whole spacetime without a time-stepping scheme. The effort raises, since the problem is of one dimension higher. But the resulting matrix is a sparse matrix and efficient solvers could be applicable in principle. Whether a bad convergence was investigated in this work with an ordinary Newton scheme, other numerical methods could succeed. In this way initial data for the Time-Evolution scheme could be improved as well. Furthermore, problems with matter could be considered. For instance, problems with neutron stars can be modelled with Regge Calculus too.

A successful running evolution scheme can be enhanced by means of the ideas coming from Finite-Element theory like adaptive and uniform refinement techniques. Since Regge Calculus can be considered to be a Finite-Element discretization with a piecewise constant metric, all ideas from the Finite-Element method can tried to be applied in Regge Calculus as well. Overcoming all mentioned problems and enhancing technical methods as to implement efficient parallelizing strategies the big picture would be to model the binary black hole inspiral. The results of this thesis are promising but the non-linear character of the Regge equations do not allow for a straightforward progress. In my humble opinion, if this field gets more attraction and more manpower develops Regge Calculus it is possible that it can compete with existing numerical methods in the future. A motivation for Regge Calculus is always given with its beautiful immediate and direct discretization of the Einstein-Hilbert action which yield a completely coordinate-free representation.

Bibliography

[AAH

[AAH$^+$03] M. Alcubierre, G. Allen, S. Husa, J. Winicour, et al. *Toward standard testbeds for numerical relativity.* arxiv:gr-qc/0305023v1 (2003).

[BER95] R. Beck, B. Erdmann, and R. Roitzsch. *KASKADE 3.0. An Object-Oriented Adaptive Finite Element Code* (1995).

[BG01] L. C. Brewin and A. P. Gentle. *On the convergence of Regge calculus to general relativity.* Institute of Physics Publishing, **18**: 517 (2001).

[BGM$^+$97] J. W. Barrett, M. Galassi, W. A. Miller, R. D. Sorkin, P. A. Tuckey, and R. M. Williams. *Parallelizable Implicit Evolution Scheme for Regge Calculus.* International Journal of Theoretical Physics, **36**(4): 815 (1997).

[BHW$^+$07] M. C. Babiuc, S. Husa, J. Winicour, et al. *Implementation of standard testbeds for numerical relativity.* arxiv:07093559v3 [gr-qc] (2007).

[BJ80] J. M. Bowen and J. W. York Jr. *Time-asymetric initial data for black holes and black-hole collisions.* Physical Review D, **21**(8): 2047 (1980).

[Bre95] L. Brewin. *The Regge Calculus is not an approximation to General Relativity.* arxiv:gr-qc/9502043v1 (1995).

[Bre97] L. Brewin. *Riemann Normal Coordinates, Smooth Lattices and Numerical Relativity.* arxiv:gr-qc/9701057v1 (1997).

[Bre98] L. Brewin. *An ADM 3+1 formalism for smooth lattice general relativity.* Class Quantum Grav, **15**: 2427 (1998).

[Bre01] L. Brewin. *Long term stable integration of a maximally sliced Schwarzschild black hole using a smooth lattice method.* arxiv:gr-qc/0107030v1 (2001).

[Bro65] C. G. Broyden. *A Class of Methods for Solving Nonlinear Simultaneous Equations.* Mathematics of Computation, **19**(92): 557 (1965).

[BS05] A. Bliss and F. E. Su. *Lower Bounds for Simplicial Covers and Triangulations of Cubes.* Discrete Comput Geom, **33**: 669 (2005).

[Car03] S. Carroll. *Spacetime and Geometry: Introduction to General Relativity* (Prentice Hall, 2003).

[Coo91] G. B. Cook. *Initial data for axisymmetric black-hole collisions.* Physical Review D, **44**(10): 2983 (1991).

[Deu06] P. Deuflhard. *Newton Methods for Nonlinear Problems* (Springer Series in Computational Mathematics, 2006).

[DH03] P. Deuflhard and A. Hohmann. *Numerical Analysis in Modern Scientific Computing – An Introduction* (Springer, 2003).

[Gal93] M. Galassi. *Lapse and shift in Regge calculus.* Physical Review D, **47**(8): 3254 (1993).

[Gen04] A. P. Gentle. *Regge calculus: a unique tool for numerical relativity.* arxiv:gr-qc/0408006v1 (2004).

[GM97] A. P. Gentle and W. A. Miller. *A fully (3+1)-D Regge calculus model of the Kasner cosmology.* arxiv:gr-qc/9706034v2 (1997).

[Gow71] R. H. Gowdy. *Gravitational Waves in Closed Universes.* Physical Review Letters, **27**(12): 826 (1971).

[GR07] C. Grossmann and H. Roos. *Numerical Treatment of Partial Differential Equations* (Springer, 2007).

[HK04] H. W. Hamber and G. Kagel. *Exact Bianchi identity in Regge gravity.* Class Quantum Grav, **21**: 5915 (2004).

[Joh94] C. Johnson. *Numerical solution of partial differential equations by the finite element method* (Cambridge University Press, 1994).

[Lei90] P. Leinen. *Ein Schneller Adaptiver Lï¿½er fr Elliptische Randwertprobleme auf Seriell- und Parallelrechnern.* PhD thesis, University of Dortmund (1990).

[Mar76] P. S. Mara. *Triangulations for the Cube.* Journal of Combinatorial Theory (A), **20**: 170 (1976).

[Mau95] J. M. Maubach. *Local Bisection Refinement for N-Simplicial Grids Generated by Reflection.* SIAM J Sci Comput, **16**(1): 210 (1995).

[Mil86] W. A. Miller. *The Geometricdynamic Content of the Regge Equations as Illuminated by the Boundary of Boundary Principle.* Foundations of Physics, **16**(2): 143 (1986).

[Mil95] M. A. Miller. *Regge calculus as a fourth-order method in numerical relativity.* Class Quantum Grav, **12**: 3037 (1995).

[Mit89] W. F. Mitchell. *A Comparison of Adaptive Refinement Techniques for Elliptic Problems.* ACM Transactions of Mathematical Software, **15**(4): 326 (1989).

[MTW73] C. W. Misner, K. S. Thorne, and J. A. Wheeler. *Gravitation* (Palgrave Macmillan, 1973).

[Muk96] A. Mukherjee. *An Adaptive Finte Element Code For Elliptic Boundary Value Problems In Three Dimensions With Applications in Numerical Relativity.* PhD thesis, Pennsylvania State University (1996).

[Pao92] C. V. Pao. *Nonlinear Parabolic and Elliptic Equations* (Plenum Press, 1992). Chapter 1.7: Monotone Method for Steady-State Problems.

[Reg] T. Regge. *Introduction to Relativity (CERN lectures).* Avaiable at http://www.dleex.com. Old typewriter document. Year of Publication unknown.

[Reg61] T. Regge. *General Relativity without Coordinates.* Nuovo Cimento, **XIX**(3): 558 (1961).

[Rus45] B. Russell. *A History of Western Philosophy* (Simon and Schuster, 1945).

[SCA$^+$08] S. Schlamminger, K. Choi, E. Adelberger, et al. *Test of the Equivalence Principle Using a Rotating Torsion Balance.* Physical Review Letters, **100**: 041101 (2008).

[Sor74] R. D. Sorkin. *Development of Simplectic Methods for the Metrical and Electromagnetic Fields.* PhD thesis, California Institute of Technology (1974).

[Sor75] R. Sorkin. *Time-evolution problem in Regge calculus.* Physical Review D, **12**(2): 385 (1975).

[Spe01] U. Sperhake. *Non-linear numerical Schemes in General Relativity.* PhD thesis, University of Southampton (2001).

[TVF02] S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++* (Cambridge University Press, 2002).

List of Figures

List of Tables

List of Algorithms

Acknowledgements

I want to thank Prof. Dr. Gerhard ZUMBUSCH for the great working conditions, supervision, support and the idea to work on the field of Regge Calculus. More general, I want to thank the Friedrich-Schiller Universität which made this thesis financally possible and gave the generic setting to be scientifically active. I also want to thank the Sonderforschungsbereich/Trangsregio 7 - Gravitational Wave Astronomy. It was a pleasure for me to work and exchange ideas in the context of a unique collection of working groups in Germany which not only employ great scientific staff but also host international luminaries. I feel deeply thankful for this opportunity and participation.

Curriculum Vitae

Completing school education and basic military training in autumn 2000, I started my scientific career in Ilmenau at the Technical University with the studies of electrical engineering. After one year, I adjusted my field of interest and started to study phyisics in Jena at the Friedrich-Schiller-Universität. In 2004, I accomplished a four-month lasting internship at the Electricity Generation Authority Thailand (EGAT). In spring 2006, I finished my diploma thesis on the subject of electronical and structural properties of calcium fluorine in the field of theoretical condensed matter physics in cooperation with the research facility *INNOVENT, Jena.* Straight after achieving the academical grade *Diplom-Physiker*, equivalent to *Master of Physics*, I started to work as a research assistant at the Institute of Applied Mathematics, Friedrich-Schiller-Universität Jena in the scientific computing group of Prof. Dr. Gerhard Zumbusch.